VYSOKOŠKOLSKÉ SKRIPTÁ

Matematicko-fyzikálna fakulta Univerzity Komenského, Bratislava

Katedra informatiky

Eduard Toman

Vybrané partie z logiky

Verzia 2.0, 12. februára 2002

BRATISLAVA 1998

Vybrané partie z logiky

Verzia 2.0, zo dňa 12. februára 2002. Neprešlo jazykovou úpravou. Sadzba systémom LaTeX 2_ε

KAPITOLA 1

Sémantika formúl logiky prvého rádu

1.1. Interpretácia formúl v logike prvého rádu

Vo výrokovej logike je interpretácia pripísanie pravdivostnej hodnoty elemetárnym formulám — logickým premenným. V logike prvého rádu musíme urobiť viac, pretože formuly sú zložitejšie objekty ako vo výrokovej logike. Na to, aby sme určili interpretáciu pre formuly logiky prvého rádu, musíme určiť predmetovú oblasť — oblasť hodnôt predmetových premenných a konštánt — *univerzum* a vhodne interpretovať funkčné a predikátové symboly, ktoré sa vyskytujú vo formule.

Stručne povedané, interpretácia formuly A logiky prvého rádu sa skladá z neprázdnej predmetovej oblasti (univerza) — množiny *indivíduí* (niekedy pre ňu používame označenie D alebo M) a z určenia hodnôt všetkých konštánt, funkčných symbolov a predikátových symbolov, vyskytujúcich sa v A. Teda každej konštante priradíme niektorý element z D, každému n-árnemu funkčnému symbolu priradíme zobrazenie z D^n do D (poznamenávame, že $D^n = \{(x_1, x_2, \ldots, x_n) \mid x_1 \in \in D, x_2 \in D, \ldots, x_n \in D\}$) a každému n-árnemu predikátovému symbolu priradíme zobrazenie $D^n \to \{0, 1\}$. Niekedy, aby sme obrátili pozornosť na D, hovoríme o interpretácii formuly na D.

Keď hľadáme „hodnotu", t.j. určujeme pravdivostnú hodnotu formuly v interpretácii na oblasti D, „$(\forall x)$" interpretujeme ako „pre všetky prvky x z D", „$(\exists x)$" ako „existuje prvok x z D". Pre každú interpretáciu formuly na oblasti D formula môže nadobudnúť pravdivostnú hodnotu 1 — *pravda* alebo 0 — *nepravda* v súlade so známymi pravidlami výrokovej logiky. V prípade kvantifikovaných výrokov je situácia obdobná.

Poznamenávame, že formula obsahujúca voľné premenné nemôže nadobudnúť pravdivostnú hodnotu. V ďalšom budeme predpokladať, že formula buď neobsahuje voľné premenné, alebo voľné premenné uvažujeme ako konštanty.

V logike prvého rádu je nekonečne veľa oblastí, teda vo všeobecnosti povedané, máme nekonečne veľa interpretácií formuly. Z toho vyplýva, že na rozdiel od výrokovej logiky, nemôžeme dokázať všeobecnú platnosť — *tautologičnosť* alebo nesplniteľnosť formuly ohodnotením formuly pre všetky možné interpretácie. Našou úlohou bude uviesť procedúry na preverenie nesplniteľnosti formúl v logike prvého rádu. Aby sme zjednodušili procedúry dôkazov, budeme pracovať s formulami, ktoré sú vyjadrené prefixovou normálnou formou, t.j. v tvare

$$(Q_1 x_1) \cdots (Q_n x_n)(M)$$

kde každé $(Q_i x_i)$ $(i = 1, \ldots, n)$ je alebo $(\forall x_i)$ alebo $(\exists x_i)$ a M je formula, neobsahujúca kvantifikátory. $(Q_1 x_1) \cdots (Q_n x_n)$ sa nazýva *prefix* a M *matica formuly*, *jadro formuly*. (Odteraz budeme M používať len na označenie matice a nie univerza, t.j. na označenie oblasti interpretácie budeme používať výlučne symbol D.) V ďalšom budemem používať označenie $A \equiv B$, ak A a B nadobúdajú rovnaké pravdivostné hodnoty pri ľubovoľnej interpretácii. Niektoré ekvivalentné dvojice formúl už poznáme; uvedieme tie, ktoré obsahujú kvantifikátory.

Nech A je formula, ktorá obsahuje voľnú premennú x (budeme to označovať $A(x)$) a nech B je formula, ktorá neobsahuje x. Potom máme nasledujúce dvojice ekvivalentných formúl, kde Q je buď „$\forall$" alebo „$\exists$":

$$(Qx)A(x) \vee B \equiv (Qx)\big(A(x) \vee B\big) \tag{1a}$$

$$(Qx)A(x) \wedge B \equiv (Qx)\big(A(x) \wedge B\big) \tag{1b}$$

$$\neg(\forall x)A(x) \equiv (\exists x)\neg A(x) \tag{2a}$$

$$\neg(\exists x)A(x) \equiv (\forall x)\neg A(x) \tag{2b}$$

Zákony (1a) a (1b) sú zrejme pravdivé, pretože B neobsahuje x, a teda môže byť uvedená do oblasti pôsobenia kvantifikátora Q. Zákony (2a), (2b) nie je ťažké dokázať: Nech I je ľubovoľná interpretácia s oblasťou D. Ak $\neg(\forall x)A(x)$ je pravdivá v I, tak $(\forall x)A(x)$ neplatí v I. To znamená, že existuje taký prvok a v D, že $A(a)$ neplatí, t.j. $\neg A(a)$ platí v I, teda $(\exists x)\neg A(x)$ platí v I. Z druhej strany, ak $\neg(\forall x)A(x)$ neplatí v I, tak $(\forall x)A(x)$ platí v I. To znamená, že $A(x)$ platí pre každý element $x \in D$, a teda $(\exists x)\neg A(x)$ neplatí v I. Pretože $\neg(\forall x)A(x)$ a $(\exists x)\neg A(x)$ vždy nadobúdajú jednu a tú istú hodnotu pre ľubovoľnú interpretáciu, tak podľa definície $\neg(\forall x)A(x) \equiv (\exists x)\neg A(x)$. Zákon (2a) je dokázaný. Analogicky môžeme dokázať zákon (2b).

Predpokladajme, že $A(x)$ a $B(x)$ sú dve formuly, ktoré obsahujú x voľne. Potom

$$(\forall x)A(x) \ \wedge \ (\forall x)B(x) \equiv (\forall x)\big(A(x) \ \wedge \ B(x)\big) \tag{3a}$$

$$(\exists x)A(x) \ \vee \ (\exists x)B(x) \equiv (\exists x)\big(A(x) \ \vee \ B(x)\big) \tag{3b}$$

t.j. kvantifikátor $\forall$ a kvantifikátor $\exists$ môžeme rozdeľovať podľa $\wedge$, resp. $\vee$. Kvantifikátor $\forall$ a kvantifikátor $\exists$ nemôžeme rozdeľovať podľa $\vee$, resp. $\wedge$, t.j.

$$(\forall x)A(x) \ \vee \ (\forall x)B(x) \not\equiv (\forall x)\big(A(x) \ \vee \ B(x)\big)$$

$$(\exists x)A(x) \ \wedge \ (\exists x)B(x) \not\equiv (\exists x)\big(A(x) \ \wedge \ B(x)\big)$$

Na overenie stačí vziať $D = \{a, b\}$ a vhodne definovať pravidovosť formúl $A(x)$ a $B(x)$.

V podobných prípadoch musíme postupovať špeciálnymi spôsobmi. Pretože každá viazaná premenná vo formule sa môže uvažovať ako miesto pre zámenu ľubovoľnej premennej, tak každú viazanú premennú x môžeme premenovať na z a formula $(\forall x)B(x)$ prejde do formuly $(\forall z)B(z)$, t.j. $(\forall x)B(x) \equiv (\forall z)B(z)$. Predpokladáme, že vyberieme premennú, ktorá sa nevyskytuje v $A(x)$. Potom

$$\begin{aligned}(\forall x)A(x) \ \vee \ (\forall x)B(x) &\equiv (\forall x)A(x) \ \vee \ (\forall z)B(z) \\ &\quad \text{(tak, že nahradíme všetky } x\text{, vyskytujúce sa v } (\forall x)B(x)\text{)} \\ &\equiv (\forall x)(\forall z)\big(A(x) \ \vee \ B(z)\big) \qquad \text{(podľa (1a))}\end{aligned}$$

Analogicky dostávame

$$\begin{aligned}(\exists x)A(x) \ \wedge \ (\exists x)B(x) &\equiv (\exists x)A(x) \ \wedge \ (\exists z)B(z) \\ &\quad \text{(tak, že nahradíme všetky } x\text{, vyskytujúce sa v } (\exists x)B(x)\text{)} \\ &\equiv (\exists x)(\exists z)\big(A(x) \ \wedge \ B(z)\big) \qquad \text{(podľa (1b))}\end{aligned}$$

Teda, pre tieto dva prípady vždy môžeme vyniesť všetky kvantifikátory vo formule naľavo. Vo všeobecnom prípade dostávame

$$(Q_1x)A(x) \ \vee \ (Q_2x)B(x) \equiv (Q_1x)(Q_2z)\big(A(x) \ \vee \ B(z)\big) \tag{4a}$$

$$(Q_3x)A(x) \ \wedge \ (Q_4x)B(x) \equiv (Q_3x)(Q_4z)\big(A(x) \ \wedge \ B(z)\big) \tag{4b}$$

kde Q_1, Q_2, Q_3 a Q_4 sú „$\forall$" alebo „$\exists$" a z nevystupuje v $A(x)$. Ak $Q_1 = Q_2 = \exists$ a $Q_3 = Q_4 = \forall$, tak netreba premenovať premennú x v $(Q_2x)B(x)$ alebo $(Q_4x)B(x)$. V tomto prípade môžeme priamo použiť formuly (3a), (3b). Ak použijeme známe zákony, môžeme každú formulu transformovať na prefixový normálny tvar. Uvedieme stručne algoritmus na takúto transformáciu:

Krok 1. Odstránenie ekvivalencií a implikácií:

$$A \leftrightarrow B \equiv (A \to B) \ \wedge \ (B \to A)$$

$$A \to B \equiv \neg A \ \vee \ B$$

Krok 2. Odstránenie dvojitej negácie a presun negácie k formule:

$$\begin{aligned}
\neg(\neg A) &\equiv A \\
\neg(A \vee B) &\equiv \neg A \wedge \neg B \\
\neg(A \wedge B) &\equiv \neg A \vee \neg B \\
\neg(\forall x)A(x) &\equiv (\exists x)\neg A(x) \\
\neg(\exists x)A(x) &\equiv (\forall x)\neg A(x)
\end{aligned}$$

Krok 3. Premenovanie viazaných premenných, ak je to nevyhnutné.

Krok 4. Použijeme zákony

$$\begin{aligned}
(Qx)A(x) \vee B &\equiv (Qx)\big(A(x) \vee B\big) \\
(Qx)A(x) \wedge B &\equiv (Qx)\big(A(x) \wedge B\big) \\
(\forall x)A(x) \wedge (\forall x)B(x) &\equiv (\forall x)\big(A(x) \wedge B(x)\big) \\
(\exists x)A(x) \vee (\exists x)B(x) &\equiv (\exists x)\big(A(x) \vee B(x)\big) \\
(Q_1x)A(x) \vee (Q_2x)B(x) &\equiv (Q_1x)(Q_2z)\big(A(x) \vee B(z)\big) \\
(Q_3x)A(x) \wedge (Q_4x)B(x) &\equiv (Q_3x)(Q_4z)\big(A(x) \wedge B(z)\big)
\end{aligned}$$

kde premenná z sa nevyskytuje vo formule $A(x)$.

Príklad 1.1.

- Vo formule $(\forall x)P(x,y)$ je premenná x viazaná a premenná y voľná.
- Vo formule $(\forall x)P(x,y) \wedge (\forall y)Q(y)$ je y aj voľná aj viazaná premenná.

Príklad 1.2.

$$\begin{aligned}
&(\forall x)(\forall y)\Big((\exists z)\big(P(x,z) \wedge P(y,z)\big) \to (\exists u)Q(x,y,u)\Big) \equiv \\
\equiv\ &(\forall x)(\forall y)\Big(\neg(\exists z)\big(P(x,z) \wedge P(y,z)\big) \vee (\exists u)Q(x,y,u)\Big) \equiv \\
\equiv\ &(\forall x)(\forall y)\Big((\forall z)\big(\neg P(x,z) \vee \neg P(y,z)\big) \vee (\exists u)Q(x,y,u)\Big) \equiv \\
\equiv\ &(\forall x)(\forall y)(\forall z)(\exists u)\big(\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,u)\big)
\end{aligned}$$

KAPITOLA 2

Herbrandova veta

Mnohé úlohy sa dajú riešiť pomocou dôkazov — odvodením teorém. Dôležitou úlohou logiky je aj štúdium mechanických precedúr hľadania dôkazov (odvodení) teorém. Poznamenávame, že hľadanie všeobecnej rozhodnuteľnej procedúry na preverenie toho, či daná formula je tautológia alebo nie je, patrí k starým úlohám. G. Leibniz (1646–1716) bol prvý, kto sa snažil nájsť takúto procedúru. Na hranici 20. storočia sa pokúšal nájsť takúto procedúru Peáno, ďalej to bola Hilbertova škola okolo roku 1920. To pokračovalo, pokým A. Church a A. M. Turing (1936) nezávisle nedokázali, že neexistuje žiadna všeobecne rozhodnuteľná procedúra, teda žiadny algoritmus, preverujúci tautologičnosť formúl v logike prvého rádu. Existujú však algoritmy nájdenia dôkazu, ktoré môžu potvrdiť, že formula je tautológia, ak skutočne je tautológiou. Pre formuly, ktoré nie sú tautológiami tieto algoritmy vo všeobecnosti povedané nezakončujú svoju činnosť. Ak berieme do úvahy výsledok Churcha a Turinga, je to to najlepšie, čo môžeme očakávať od algoritmu nájdenia dôkazu.

Významný vklad do teórie automatického dokazovania teorém vniesol Herbrand (1930). Herbrand rozpracoval algoritmus nájdenia interpretácie, ktorá vyvracia danú formulu. Ak však daná formula je tautológia, potom neexistuje taká interpretácia a algoritmus zakončuje svoju činnosť za konečný počet krokov. Herbrandova metóda tvorí základ automatických procedúr pre nájdenie dôkazu (odvodenia) teorém.

Gilmore (1960) ako jeden z prvých realizoval procedúru navrhnutú Herbrandom na počítači. Formula je tautológiou práve vtedy, keď jej negácia je nesplniteľná. Jeho program je určený na odhalenie nesplniteľnosti negácie danej formuly. Počas uskutočňovania jeho programu sa indukujú logické formuly, u ktorých sa pravidelne preveruje nesplniteľnosť. Ak je negácia danej formuly nesplniteľná, tak program nakoniec zaznamená tento fakt. Gilmoreho program efektívne pracoval pre dôkazy jednoduchých formúl, no stretol sa s veľkými ťažkosťami pri dôkazoch zložitejších formúl logiky prvého rádu. Pozorné štúdium jeho programu ukázalo, že metóda preverenia nesplniteľnosti logických formúl je neefektívna. Gilmoreho metódu zlepšil Davis s Putnamom (1960) niekoľko mesiacov po tom, čo bola opublikovaná, no rovnako ich zlepšenie bolo ešte nedostatočné. Mnohé tautológie logiky prvého rádu sa ešte nedajú na počítači dokázať za rozumný čas.

Hlavný skok urobil Robinson (1965–1968), ktorý zaviedol *metódu rezolvent*. Procedúra nájdenia dôkazu rezolvenčnou metódou je omnoho efektívnejšia ako ľubovoľná procedúra opísaná predtým. Po zavedení rezolvenčnej metódy boli rozpracované aj ďalšie stratégie za účelom zvýšenia jej efektívnosti. My budeme rozoberať dôkaz Herbrandovej vety a vyložíme metódu rezolvent.

2.1. Skolemovské štandardné formy

Procedúry nájdenia dôkazu Herbrandovou metódou alebo metódou rezolvent sú v skutočnosti procedúrami nájdenia vyvrátenia, t.j. namiesto dôkazu tautologičnosti formuly sa dokazuje, že jej negácia je nesplniteľná. To je len otázka vhodnosti — pri používaní procedúr vyvrátenia sa všeobecnosť nestráca. Procedúry vyvrátenia aplikujeme na *štandardnú formu* formuly, ktorú zaviedli Davis a Putnam. V podstate Davis a Putnam použili nasledujúce tvrdenia:

(1) Formula logiky prvého rádu môže byť vyjadrená v prefixovej normálnej forme, v ktorej matica neobsahuje žiadne kvantifikátory a prefix je postupnosť kvantifikátorov.
(2) Pretože matica neobsahuje kvantifikátory, môže byť vyjadrená v konjunktívnej normálnej forme.
(3) Zachovávajúc nesplniteľnosť formúl, môžeme v nej eliminovať existenčné kvantifikátory pomocou použitia *skolemovských funkcií*.

Nech sa formula A nachádza v prefixovom normálnom tvare $(Q_1x_1)\cdots(Q_nx_n)M$, kde M je konjunktívna normálna forma. Predpokladajme, že Q_r je existenčný kvantifikátor v prefixe

$$(Q_1x_1)\cdots(Q_nx_n) \qquad (1 \leq r \leq n)$$

Ak žiaden kvantifikátor „$\forall$" nestojí v prefixe vľavo od Q_r, vyberieme konštantu c, rôznu od iných konštánt, vyskytujúcich sa v M, zameníme všetky výskyty premennej x_r, vyskytujúce sa v M konštantou c a vyčiarkneme (Q_rx_r) z prefixu. Ak $Q_{s_1},\ldots,Q_{s_m}$ je zoznam všetkých kvantifikátorov „$\forall$", ktoré vystupujú vľavo od Q_r $(1 \leq s_1 < s_2 < \cdots < s_m < r)$, vyberieme nový m-miestny funkčný symbol f rôzny od iných funkčných symbolov, zameníme všetky x_r za $f(x_{s_1},\ldots,x_{s_m})$ a vyčiarkneme (Q_rx_r) z prefixu. Tento proces zopakujeme pre všetky kvantifikátory „$\exists$" v prefixe: posledná z dosiahnutých formúl je *skolemovská štandardná forma* — skrátene *štandardná forma* formuly A. Konštanty a funkcie, ktoré sme použili na zámenu premenných kvantifikátora existencie, sa nazývajú *skolemovské funkcie*.

Príklad 2.1. Nájdite štandardnú formu formuly

$$(\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(x,y,z,u,v,w)$$

Riešenie: V tejto formule vľavo od $(\exists x)$ niet všeobecných kvantifikátorov, vľavo od $(\exists u)$ sú $(\forall y)$ a $(\forall z)$ a vľavo od $(\exists w)$ sú $(\forall y)$, $(\forall z)$ a $(\forall v)$. Z toho vyplýva, že premennú x zameníme konštantou a, premennú u binárnou funkciou $f(y,z)$ a premennú w ternárnou funkciou $g(y,z,v)$. Týmto spôsobom dostávame nasledujúcu štandardnú formu vyššie uvedenej formuly:

$$(\forall y)(\forall z)(\forall v)P(a,y,z,f(y,z),v,g(y,z,v))$$

Príklad 2.2. Nájdite štandardnú formu pre formulu

$$(\forall x)(\exists y)(\exists z)\Big(\big(\neg P(x,y) \ \wedge \ Q(x,z)\big) \ \vee \ R(x,y,z)\Big)$$

Riešenie: Najprv napíšeme maticu v konjunktívnej normálnej forme:

$$(\forall x)(\exists y)(\exists z)\Big(\big(\neg P(x,y) \ \vee \ R(x,y,z)\big) \ \wedge \ \big(Q(x,z) \ \vee \ R(x,y,z)\big)\Big)$$

Pred $(\exists y)$ aj $(\exists z)$ je $(\forall x)$, preto premenné y, resp. z zamieňame unárnymi funkciami $f(x)$, resp. $g(x)$. Takým spôsosbom dostávame štandardnú formu:

$$(\forall x)\Big(\big(\neg P(x,f(x)) \ \vee \ R(x,f(x),g(x))\big) \ \wedge \ \big(Q(x,g(x)) \ \vee \ R(x,f(x),g(x))\big)\Big)$$

Definícia 2.3. *Klauzula* je disjunkcia literálov. (Pod *literálom* rozumieme logickú premennú alebo jej negáciu.)

Niekedy je užitočné uvažovať množinu literálov ako synonymum klauzuly. Napr., $P \vee Q \vee \neg R = \{P, Q, \neg R\}$. Jednoliterálna klauzula sa nazýva *jednotková klauzula*. Keď klauzula neobsahuje žiadne literály, budeme ju nazývať *prázdnou klauzulou*. Pretože prázdna klauzula neobsahuje žiadne literály, ktoré by mohli byť pravdivé pri akejkoľvek interpretácii, tak prázdna klauzula je vždy nepravdivá. Prázdnu klauzulu označujeme „$\square$".

Disjunkcie $\neg P(x,f(x)) \vee R(x,f(x).g(x))$ a $Q(x,g(x)) \vee R(x,f(x),g(x))$ v štandardnej forme z príkladu 2.2 sú klauzuly. Predpokladáme, že množina klauzúl S je konjunkcia všetkých klauzúl z S, kde každá premenná v S je viazaná veľkým kvantifikátorom. Vďaka tejto dohode, štandardná forma môže byť prosto vyjadrená množinou klauzúl. Napr., štandardná forma z príkladu 2.2 môže byť vyjadrená množinou $\{\neg P(x,f(x)) \vee R(x,f(x),g(x)), Q(x,g(x)) \vee R(x,f(x),g(x))\}$

V nasledujúcej vete dokážeme, že môžeme eliminovať existenčné kvantifikátory a pritom zachovávame nesplniteľnosť formuly.

Veta 2.4. Nech S je množina klauzúl, ktoré tvoria štandardnú formu klauzuly A. Potom A nie je splniteľná práve vtedy, keď S nie je splniteľná.

Dôkaz: ▷ Bez ujmy na všeobecnosti môžeme predpokladať, že A je vyjadrená v prenexnej normálnej forme, t.j. $A = (Q_1x_1)\cdots(Q_nx_n)M[x_1,\ldots,x_n]$. (Používame zápis $M[x_1,\ldots,x_n]$, aby sme ukázali, že jadro M obsahuje premenné $x_1,\ldots,x_n$. Jadro niekedy nazývame aj matica). Nech Q_r je prvý existenčný kvantifikátor. Nech

$$A_1 = (\forall x_1)\cdots(\forall x_{r-1})(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M[x_1,\ldots,x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$$

kde f je skolemovská funkcia, ktorá zodpovedá x_r $(1 \leq r \leq n)$. Chceme dokázať, že A nie je splniteľná práve vtedy, keď A_1 nie je splniteľná.

Predpokladajme, že A nie je splniteľná. Keby A_1 bola splniteľná, tak by existovala taká interpretácia I, že A_1 platí v I (I vyhovuje A_1), t.j. pre všetky $x_1,\ldots,x_{r-1}$ existuje aspoň jeden element (je to práve element $f(x_1,\ldots,x_{r-1})$), pre ktorý je

$$(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M[x_1,\ldots,x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$$

je splnená (pravdivá) v I. Takým spôsobom je A splnená v I, čo je v spore s predpokladom, že A nie je splniteľná. Teda ani A_1 nemôže byť splniteľná.

Predpokladajme teraz, že A je splniteľná. Potom existuje taká interpretácia I na oblasti D, že I vyhovuje A, t.j. pre všetky $x_1,\ldots,x_{r-1}$ existuje taký element x_r, že

$$(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M[x_1,\ldots,x_{r-1},x_r,x_{r+1},\ldots,x_n]$$

je splnená v I. Rozšírime interpretáciu I tým, že pridáme funkciu f, ktorá zobrazuje $(x_1,\ldots,x_{r-1})$ na x_r pre všetky $x_1,\ldots,x_{r-1} \in D$, t.j. $f(x_1,\ldots,x_{r-1}) = x_r$. Označme toto rozšírenie ako I'. Je zrejmé, že pre všetky $x_1,\ldots,x_{r-1}$ je

$$(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M[x_1,\ldots,x_{r-1},f(x_1,\ldots,x_{r-1}),x_{r+1},\ldots,x_n]$$

splnená v I', t.j. A_1 je splnená v I', čo je v spore s predpokladom, že A_1 nie je splniteľná. A teda A nemôže byť splniteľná.

Predpokladajme teraz, že A obsahuje m existenčných kvantifikátorov. Nech $A_0 = A$. Nech A_k dostaneme z A_{k-1} zámenou prvého existenčného kvantifikátora v A_{k-1} skolemovskou funkciou $k = 1,2,\ldots,m$. Je zrejmé, že $S = A_m$. Ak použijeme tie isté úvahy ako vyššie, môžeme dokázať, že A_{k-1} nie je splniteľná práve vtedy, ak A_k nie je splniteľná $(k = 1,2,\ldots,m)$, a teda môžeme urobiť záver: A nie je splniteľná práve vtedy, keď S nie je splniteľná, čo bolo treba dokázať. ◁

Nech S je štandardná forma formuly A. Ak A nie je splniteľná, tak podľa vety 2.4 je $A \equiv S$. Ak A je splniteľná, tak vo všeobecnosti A nie je ekvivalentná s S.

Napríklad, nech A je $(\exists x)P(x)$ a S je $P(a)$. Je zrejmé, že S je štandardná forma formuly A. Nech I je nasledujúca interpretácia:

- oblasť $D = \{1,2\}$
- hodnoty pre a: 1
- hodnoty pre P: $P(1)$ — nepravda, $P(2)$ — pravda

Formula $(\exists x)P(x)$ je splnená v interpretácii I, no S nie je splnená v I, teda $A \not\equiv S$.

Poznamenávame, že formula môže mať viacej ako jednu štandardnú formu. Kvôli jednoduchosti, keď transformujeme formulu A na štandardnú formu S, zamieňame existenčné kvantifikátory skolemovskými funkciami tak jednoduchými, ako sa to dá. Ďalej, ak máme $A = A_1 \wedge A_2 \wedge \cdots \wedge \wedge A_n$, môžeme oddelene dostať množinu klauzúl S_i, kde každé S_i vyjadruje štandardnú formu A_i $(i = 1,2,\ldots,n)$. Potom nech $S = S_1 \cup \cdots \cup S_n$. Pomocou úvah podobných tým, ktoré sme použili vo vete 2.4, nie je ťažké vidieť, že A nie je splniteľná práve vtedy, keď S nie je splniteľná.

Príklad 2.5. V tomto príklade ukážeme, ako je možné vyjadriť nasledujúcu vetu v štandardnej forme: „Ak $x \circ x = e$ pre všetky x v grupe G, tak G je komutatívna." Pritom „$\circ$" je binárna operácia a e jednotka v grupe G.

Riešenie: Spočiatku budeme túto vetu formalizovať a potom vyjadríme negáciu tejto vety množinou klauzúl. Vieme, že grupa G vyhovuje nasledujúcim štyrom axiómam:

(A_1): $x, y \in G$ implikuje $x \circ y \in G$ (vlastnosť uzavretosti)
(A_2): $x, y, z \in G$ implikuje $x \circ (y \circ z) = (x \circ y) \circ z$ (vlastnosť asociatívnosti)

(A_3): $x \circ e = e \circ x$ pre všetky $x \in G$ (vlastnosť existencie jednotkového prvku)
(A_4): pre každé $x \in G$ existuje prvok $x^{-1} \in G$ taký, že $x \circ x^{-1} = x^{-1} \circ x = e$ (vlastnosť existencie inverzného prvku).

Nech $P(x, y, z)$ označuje $x \circ y = z$ a $i(x) = x^{-1}$. Potom vyššie uvedené axiómy nadobúdajú tvar:

(A'_1): $(\forall x)(\forall y)(\exists z)P(x, y, z)$
(A'_2): $(\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)\Big(\big(P(x, y, u) \wedge P(y, z, v) \wedge P(u, z, w)\big) \to P(x, v, w)\Big) \wedge$
$\wedge\ (\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)\Big(\big(P(x, y, u) \wedge P(y, z, v) \wedge P(x, v, w)\big) \to P(u, z, w)\Big)$
(A'_3): $(\forall x)P(x, e, x) \wedge (\forall x)P(e, x, x)$
(A'_4): $(\forall x)P(x, i(x), e) \wedge (\forall x)P(i(x), x, e)$

Záver vety je nasledovný:

(B): Ak $x \circ x = e$ pre všetky $x \in G$, tak G je komutatívna, t.j. $u \circ v = v \circ u$ pre všetky $u, v \in G$.

Tvrdenie B môže byť vyjadrené formulou:

(B'): $(\forall x)P(x, x, e) \to \Big((\forall u)(\forall v)(\forall w)\big(P(u, v, w) \to P(v, u, w)\big)\Big)$

Teda celá veta je vyjadrená formulou $F = A'_1 \wedge \cdots \wedge A'_4 \to B'$. Takýmto spôsobom

$$\neg F = A'_1 \wedge A'_2 \wedge A'_3 \wedge A'_4 \wedge \neg B'$$

Aby sme získali množinu klauzúl pre $\neg F$, najprv získame množinu klauzúl S_i pre každú axiómu A'_i $(i = 1, 2, 3, 4)$ nasledujúcim spôsobom:

$$\begin{aligned}
S'_1 &= \{P(x, y, f(x, y))\} \\
S'_2 &= \{\neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(u, z, w) \vee P(x, v, w), \\
&\qquad \neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(x, v, w) \vee P(u, z, w)\} \\
S'_3 &= \{P(x, e, x), P(e, x, x)\} \\
S'_4 &= \{P(x, i(x), e), P(i(x), x, e)\}
\end{aligned}$$

Platí

$$\begin{aligned}
\neg B' &= \neg\Big((\forall x)P(x, x, e) \to \Big((\forall u)(\forall v)(\forall w)\big(P(u, v, w) \to P(v, u, w)\big)\Big)\Big) = \\
&= \neg\Big(\neg(\forall x)P(x, x, e) \vee \Big((\forall u)(\forall v)(\forall w)\big(\neg P(u, v, w) \vee P(v, u, w)\big)\Big)\Big) = \\
&= (\forall x)P(x, x, e) \wedge \neg\Big((\forall u)(\forall v)(\forall w)\big(\neg P(u, v, w) \vee P(v, u, w)\big)\Big) = \\
&= (\forall x)P(x, x, e) \wedge (\exists u)(\exists v)(\exists w)\big(P(u, v, w) \wedge \neg P(v, u, w)\big)
\end{aligned}$$

Preto množina klauzúl pre $\neg B'$ je nasledujúca:

$$T = \{P(x, x, e), P(a, b, c), \neg P(b, a, c)\}$$

Preto množina $S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup T$ je množina, ktorá sa skladá z nasledujúcich klauzúl

(1) $P(x, y, f(x, y))$
(2) $\neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(u, z, w) \vee P(x, v, w)$
(3) $\neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(x, v, w) \vee P(u, z, w)$
(4) $P(x, e, x)$
(5) $P(e, x, x)$
(6) $P(i(x), x, e)$
(7) $P(x, i(x), e)$
(8) $P(x, x, e)$
(9) $P(a, b, c)$
(10) $\neg P(b, a, c)$

V príklade 2.5 sme ukázali, ako dostaneme množinu klauzúl S pre formulu $\neg F$. Zo známych tvrdení vieme, že F je tautológia práve vtedy, keď S nie je splniteľná. Ako sme už povedali, dôkazy toho, že formula je tautológia budeme prevádzať na to, že jej negácia nie je splniteľná. Preto od tohto miesta budeme predpokladať, že na vstupe procedúry uvažovaného dôkazu vždy stojí množina klauzúl (taká, ako je S, ktoré sme dostali v príklade 2.5). Ďalej používame pre množinu klauzúl termíny „nie je splniteľná" („splniteľná"), niekedy aj „protirečivá", „sporná" („neprotirečivá", „nie je sporná").

2.2. Herbrandovské univerzum množiny klauzúl

Podľa definície množina klauzúl nie je splniteľná práve vtedy, ak je nepravdivá pri všetkých interpretáciách na všetkých oblastiach. Pretože nie je vhodné a ani nie je užitočné skúmať všetky interpretácie na všetkých oblastiach, bolo by dobré, ak by sme mohli fixovať jednu špeciálnu oblasť H a pre S urobiť záver, že S nie je splniteľná práve vtedy, keď S nie je pravdivá pri žiadnej interpretácii na tejto oblasti. Na šťastie taká oblasť existuje. Nazývame ju *Herbrandovské univerzum* množiny S a definujeme nasledujúcim spôsobom.

Definícia 2.6. Nech H_0 je množina konštánt, ktoré sa vyskytujú v S. Ak sa žiadna konštanta nevyskytuje v S, tak H_0 sa skladá z jednej konštanty, povedzme $H_0 = \{a\}$. Pre $i = 0, 1, 2, \ldots$ je H_{i+1} zjednotením H_i a množiny všetkých termov tvaru $f^{(n)}(t_1, \ldots, t_n)$ pre každé n a všetky $f^{(n)}$, ktoré sa vyskytujú v S, kde $t_j \in H_i$ $(j = 1, 2, \ldots, n)$. Potom každé H_i nazývame množinou konštánt i-tej úrovne pre S a H_∞ nazývame *Herbrandovo univerzum* pre S.

Príklad 2.7. Nech $S = \{P(a), \neg P(x) \vee \neg P(f(x))\}$. Potom

$$\begin{aligned} H_0 &= \{a\} \\ H_1 &= \{a, f(a)\} \\ H_2 &= \{a, f(a), f(f(a))\} \\ &\vdots \\ H_\infty &= \{a, f(a), f(f(a)), \ldots\} \end{aligned}$$

Príklad 2.8. Nech $S = \{P(x) \vee Q(x), R(z), T(y) \vee \neg W(y)\}$. Pretože v S neexistujú žiadne konštanty, kladieme $H_0 = \{a\}$. Pretože v S neexistujú ani žiadne funkčné symboly, tak

$$H = H_0 = H_1 = \cdots = \{a\}$$

Príklad 2.9. Nech $S = \{P(f(x), a, g(y), b)\}$. Potom

$$\begin{aligned} H_0 &= \{a, b\} \\ H_1 &= \{a, b, f(a), f(b), g(a), g(b)\} \\ H_2 &= \{a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)), f(g(b)), \\ &\quad g(f(a)), g(f(b)), g(g(a)), g(g(b))\} \end{aligned}$$

V dôsledku toho, čo bolo povedané, pod *výrazom* budeme chápať term, množinu termov, množinu atomárnych formúl, literál, klauzulu, či množinu klauzúl. Ak sa vo výraze nevyskytujú žiadne premenné, aby sme zdôraznili túto skutočnosť, niekedy tento výraz nazývame *základnou inštanciou*. Takýmto spôsobom môžeme používať pojmy základný term, základný atóm, základný literál, základná klauzula, aby sme zdôraznili, že žiadne premenné sa nevyskytujú v zodpovedajúcich výrazoch.

Podvýrazom výrazu E je výraz, ktorý sa vyskytuje v E.

Definícia 2.10. Nech S je množina klauzúl. Potom množina atómov tvaru $P^{(n)}(t_1, \ldots, t_n)$ pre všetky n-árne predikáty $P^{(n)}$, ktoré sa vyskytujú v S a $t_1, \ldots, t_n$ sú elementy Herbrandovského univerza pre S, sa nazýva *Herbrandovskou bázou* pre S.

Definícia 2.11. Základná inštancia klauzuly C z množiny klauzúl S je klauzula, ktorú dostaneme zámenou premenných v C prvkami Herbrandovského univerza pre S.

Príklad 2.12. Nech $S = \{P(x), Q(f(y)) \vee R(y)\}$, t.j. $C = P(x)$ je klauzula v S a $H = \{a, f(a), f(f(a)), \dots\}$ je Herbrandovské univerzum v S. Potom $P(a)$ a $P(f(f(a)))$ sú základné inštancie C.

Uvažujme teraz interpretácie nad Herbrandovským univerzom. Nech S je množina klauzúl. Ako sme už povedali, interpretácia nad Herbrandovským univerzom množiny S je určená hodnotami konštánt, funkčných symbolov a predikátových symbolov, ktoré sa vyskytujú v S. Ďalej budeme definovať špeciálnu interpretáciu nad Herbrandovským univerzom, ktorú budeme nazývať *H-interpretáciou* množiny S.

Definícia 2.13. Nech S je množina klauzúl, H Herbrandovské univerzum pre S a I interpretácia S nad H. Hovoríme, že I je *H-interpretácia* množiny S, ak vyhovuje nasledujúcim podmienkam:

- I zobrazuje všetky konštanty z S na seba, t.j. konštante $a_i \in S$ priradí $a_i \in H$.
- Nech $f^{(n)}$ je n-árny funkčný symbol a $h_1, \dots, h_n$ sú elementy H. V I znakom $f^{(n)}$ označujeme funkciu, ktorá zobrazuje element $(h_1, \dots, h_n)$ z H^n na element $f^{(n)}(h_1, \dots, h_n)$ z H.

Poznamenávame, že nekladieme žiadne ohraničenia pri interpretácii a určovaní hodnoty ľubovoľného n-árneho predikátového symbolu z S. Nech $A = \{A_1, A_2, \dots, A_n, \dots\}$ je Herbrandovská báza množiny S. H-interpretáciu I je vhodné vyjadriť v tvare

$$I = \{m_1, m_2, \dots, m_n, \dots\}$$

kde m_j je buď A_j alebo $\neg A_j$ pre $j = 1, 2, \dots$ Zmysel tejto množiny je v tom, že ak m_j je A_j, tak atómu A_j je priradená hodnota „pravda" a v opačnom prípade hodnota „nepravda".

Príklad 2.14. Uvažujme množinu $S = \{P(x) \vee Q(x), R(f(y))\}$. Herbrandovské univerzum H pre S je $H = \{a, f(a), f(f(a)), \dots\}$. V S sa vyskytujú predikátové symboly: P, Q a R. Z toho vyplýva, že Herbrandovská báza pre S je $A = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \dots\}$. Niektoré H-interpretácie množiny S sú

$$I_1^* = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \dots\}$$
$$I_2^* = \{\neg P(a), \neg Q(a), \neg R(a), \neg P(f(a)), \neg Q(f(a)), \neg R(f(a)), \dots\}$$
$$I_3^* = \{P(a), Q(a), \neg R(a), P(f(a)), Q(f(a)), \neg R(f(a)), \dots\}$$

Interpretáciu množiny klauzúl S nie je nutné zadávať nad Herbrandovským univerzom, t.j. interpretácia nemusí byť H-interpretáciou. Nech napr. $S = \{P(x), Q(y, f(y, a))\}$. Potom je možná nasledujúca interpretácia nad oblasťou $D = \{1, 2\}$ uvedená v tabuľke 2.1. Pre takúto interpretáciu

a	$f(1,1)$	$f(1,2)$	$f(2,1)$	$f(2,2)$
2	1	2	2	1

$P(1)$	$P(2)$	$Q(1,1)$	$Q(1,2)$	$Q(2,1)$	$Q(2,2)$
platí	neplatí	neplatí	platí	neplatí	platí

TABUĽKA 2.1. Interpretácia na oblasti $D = \{1, 2\}$

môžeme určiť H-interpretáciu I^*, zodpovedajúcu I. Ilustrujeme to na tom istom príklade. Najprv nájdeme Herbrandovskú bázu pre S:

$$A = \{P(a), Q(a, a), P(f(a, a)), Q(a, f(a, a)), Q(f(a, a), a), \dots\}$$

Potom ohodnotíme každý člen množiny A tým, že použijeme tabuľku hodnôt 2.1:

$$\begin{aligned} P(a) &= P(2) = \text{neplatí} \\ Q(a,a) &= Q(2,2) = \text{platí} \\ P(f(a,a)) &= P(f(2,2)) = P(1) = \text{platí} \\ Q(a,f(a,a)) &= Q(2,f(2,2)) = Q(2,1) = \text{neplatí} \\ Q(f(a,a),a) &= Q(f(2,2),2) = Q(1,2) = \text{platí} \\ Q(f(a,a),f(a,a)) &= Q(f(2,2),f(2,2)) = Q(1,1) = \text{neplatí} \end{aligned}$$

Dôsledkom toho H-interpretácia I^* zodpovedajúca I je

$$I^* = \{\neg P(a), Q(a,a), P(f(a,a)), \neg Q(a,f(a,a)), Q(f(a,a),a), \neg Q(f(a,a),f(a,a)), \dots\}$$

Ak v S nie sú konštanty, tak element a, ktorý sme použili na to, aby sme mohli začať Herbrandovské univerzum, môžeme zobraziť na ľubovoľný element v oblasti D. V prípade, že oblasť D má viacej ako jeden prvok, tak existuje viac ako jedna H-interpretácia zodpovedajúca I. Nech napr. $S = \{P(x), Q(y,(f(y,z))$ a nech je pre S vybraná interpretácia na oblasti $D = \{1,2\}$ podľa tabuľky 2.2.

$f(1,1)$	$f(1,2)$	$f(2,1)$	$f(2,2)$
1	2	2	1

$P(1)$	$P(2)$	$Q(1,1)$	$Q(1,2)$	$Q(2,1)$	$Q(2,2)$
platí	neplatí	neplatí	platí	neplatí	neplatí

TABUĽKA 2.2. Interpretácia na oblasti $D = \{1,2\}$

Potom interpretácii I budú zodpovedať dve H-interpretácie:

- $I_1^* = \{P(a), \neg Q(a,a), P(f(a,a)), \neg Q(a,f(a,a)), \neg Q(f(a,a),a), \neg Q(f(a,a),f(a,a)), \dots\}$, ak $a = 1$
- $I_2^* = \{\neg P(a), \neg Q(a,a), P(f(a,a)), \neg Q(a,f(a,a)), \neg Q(f(a,a),a), \neg Q(f(a,a),f(a,a)), \dots\}$, ak $a = 2$

Teraz môžeme sformulovať vyššie uvedené pojmy nasledujúcim spôsobom.

Definícia 2.15. Nech I je interpretácia pre S na oblasti D. H-interpretáciou I^* *zodpovedajúcou* I je interpretácia, ktorá vyhovuje nasledujúcej podmienke: Nech $h_1, \dots, h_n$ sú elementy Herbrandovského univerza. Nech sa každé h_i zobrazuje v interpretácii I na niektoré $d_i \in D$. Ak $P^{(n)}(d_1, \dots, d_n)$ dostáva v interpretácii I hodnotu pravda (resp. nepravda), tak $P^{(n)}(h_1, \dots, h_n)$ taktiež dostáva hodnotu pravda (resp. nepravda) v interpretácii I^*.

V skutočnosti nie je ťažké dokázať nasledujúcu lemu:

Lema 2.16. Ak interpretácia I na niektorej oblasti D vyhovuje množine klauzúl S, tak ľubovoľná z H-interpretácií I^*, ktorá zodpovedá I, taktiež vyhovuje S.

Dôkaz: ▷ Nech $S = \{C_1, C_2, \dots, C_n\}$ je množina klauzúl, $C_i = L_{i_1} \vee L_{i_2} \vee \dots \vee L_{i_{r_i}}$ $(i = 1, \dots, n)$. Predpokladajme, že interpretácia I na niektorej oblasti D vyhovuje množine klauzúl S. To znamená, že každá z klauzúl C_i nadobúda hodnotu „pravda", t.j. v každom C_i existuje aspoň jeden literál L_{i_j} tvaru $P^{(n)}(d_1, \dots, d_n)$, ktorý je pravdivý. Nech $h_1, \dots, h_n$ sú prvky H-univerza a každé h_i sa zobrazuje na d_i v oblasti D $(i = 1, \dots, n)$. Na základe uvedeného aj literál $P^{(n)}(h_1, \dots, h_n)$ je pravdivý v interpretácii I^*, t.j. v ľubovoľnej interpretácii zodpovedajúcej I. ◁

Veta 2.17. Množina klauzúl S nie je splniteľná práve vtedy, keď S je nepravdivá pri všetkých H-interpretáciách v S.

Dôkaz: ▷ ($\Rightarrow$) Prvá polovica uvedenej vety je zrejmá, pretože podľa definície je S nesplniteľná práve vtedy, keď S je nepravdivá pri všetkých interpretáciách na ľubovoľnej oblasti.

($\Leftarrow$) Aby sme dokázali druhú polovicu predloženej vety, predpokladajme, že S je nepravdivá pri všetkých H-interpretáciách v S. Predpokadajme, že S je splniteľná. Potom existuje taká interpretácia I na niektorej oblasti D, že S je pravdivá pri I. Nech I^* je H-interpretácia zodpovedajúca I. V súlade s lemou 2.16, S je pravdivá pri I^* a to je v spore s predpokladom, že S nie je pravdivá pri všetkých H-interpretáciách v S. Teda S nie je splniteľná, čo bolo treba dokázať. ◁

Takým spôsobom sme dosiahli cieľ, ktorý sme si vytýčili na začiatku tejto časti, teda nevyhnutne nám treba uvažovať len interpretácie nad H-univerzom, t.j. H-interpretácie, na preverenie toho, či je splniteľná množina klauzúl alebo nie je. Poznamenávame, že ak odteraz budeme uvažovať interpretáciu, tak máme na mysli H-interpretáciu.

Nech $\emptyset$ označuje prázdnu množinu. Každý z nasledujúcich výrokov je zrejmý:

(1) Základná inštancia C' klauzuly C je splniteľná v interpretácii I práve vtedy, keď existuje základný literál $L' \in C'$ taký, že L' je taktiež v I, t.j.

$$C' \cap I \neq \emptyset, \qquad L' \in I = \{m_1, m_2, \ldots, m_j, \ldots\}$$

(2) Klauzula C je splnená v interpretácii I práve vtedy, keď každá jej základná inštancia C' je splnená v interpretácii I.

(3) Klauzula C je odmietnutá (vyvrátená) interpretáciou I práve vtedy, keď existuje aspoň jedna taká základná inštancia C' pre C, že C' nie je splnená v I.

(4) Množina klauzúl S nie je splniteľná práve vtedy, keď pre každú interpretáciu I existuje aspoň jedna taká základná inštancia C' niektorej klauzuly C v S, že C' nie je splnená v I.

Príklad 2.18. (a) Uvažujme klauzulu $C = \neg P(x) \vee Q(f(x))$. Nech I_1, I_2, I_3 sú definované nasledujúcim spôsobom:

$$I_1 = \{\neg P(a), \neg Q(a), \neg P(f(a)), \neg Q(f(a)), \neg P(f(f(a))), \neg Q(f(f(a))), \ldots\}$$
$$I_2 = \{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))), \ldots\}$$
$$I_3 = \{P(a), \neg Q(a), P(f(a)), \neg Q(f(a)), P(f(f(a))), \neg Q(f(f(a))), \ldots\}$$

Môžeme sa ľahko presvedčiť, že C je splnená v interpretáciách I_1 a I_2, no zamietnutá v interpretácii I_3.

(b) Uvažujme množinu $S = \{P(x), \neg P(a)\}$. Existujú dve H-interpretácie $I_1 = \{P(a)\}$ a $I_2 = \{\neg P(a)\}$. S je zamietnutá oboma interpretáciami. Z toho vyplýva, že S nie je splniteľná.

2.3. Sémantické stromy

Po zavedení H-univerza uvažujme *sémantické stromy*. Ako uvidíme neskôr, nájdenie dôkazu pre množinu klauzúl je ekvivalentné zostrojeniu sémantického stromu pre množinu klauzúl.

Definícia 2.19. Nech A je atóm. Hovoríme, že dva literály A a $\neg A$ sú navzájom *kontrárne*. Množina $\{A, \neg A\}$ sa nazýva *kontrárnou dvojicou*.

Poznamenávame, že ak klauzula obsahuje kontrárnu dvojicu, tak je tautológiou. Pri použití pojmu „tautológia" máme na mysli klauzulu, ktorá je tautológiou.

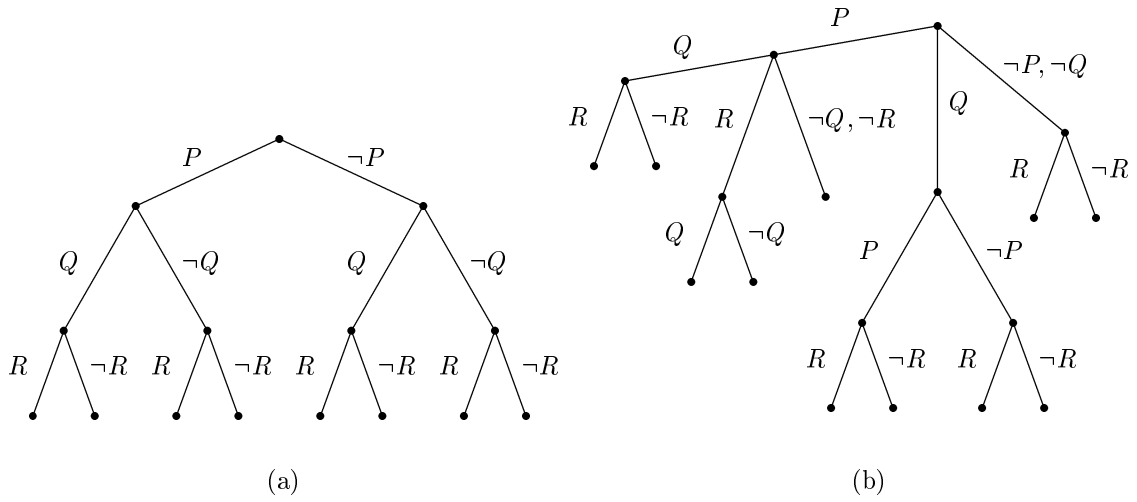
Definícia 2.20. Nech S je množina klauzúl a A jej Herbrandovská báza. Sémantický strom pre S je dole rastúci strom, v ktorom je každej hrane pripísaná množina atómov alebo negácií atómov z A takým spôsobom, že:

- Z každého vrchola v vychádza konečný počet hrán $l_1, \ldots, l_n$. Nech Q_i je konjunkcia všetkých literálov, pripísaných k l_i $(i = 1, 2, \ldots, n)$. Potom $Q_1 \vee Q_2 \vee \cdots \vee Q_n$ je všeobecne platná logická formula.

- Nech pre každý vrchol v je $I(v)$ zjednotenie všetkých množín literálov, ktoré sú pripísané hranám vetvy, ktorá vedie k v. Potom $I(v)$ neobsahuje kontrárne dvojice.

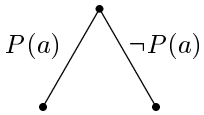
Definícia 2.21. Nech $A = \{A_1, A_2, \dots, A_n, \dots\}$ je Herbrandovská báza množiny S. Hovoríme, že sémantický strom pre S je *úplný*, ak pre každé i $(i = 1, 2, \dots)$ a každý koncový vrchol v sémantického stromu (t.j. vrchol, z ktorého nevychádzajú žiadne hrany) $I(v)$ obsahuje buď A_i alebo $\neg A_i$.

Príklad 2.22. Nech $A = \{P, Q, R\}$ je Herbrandovská báza množiny S. Potom každý z dvoch stromov na obrázku 2.1 je úplný sémantický strom pre S.



Obrázok 2.1. Stromy k príkladu 2.22

Príklad 2.23. Uvažujme $S = \{P(x), P(a)\}$. Herbrandovská báza množiny S je $\{P(a)\}$. Úplný sémantický strom pre S je na obrázku 2.2.

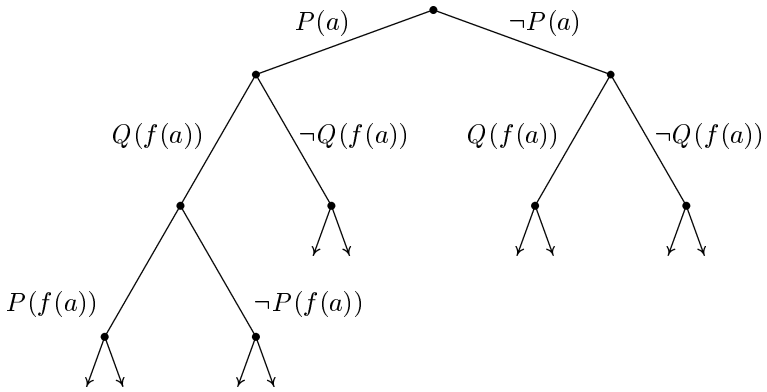


Obrázok 2.2. Strom k príkladu 2.23

Príklad 2.24. Uvažujme $S = \{P(x), Q(f(x))\}$. Herbrandovská báza množiny S je $\{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))), \dots\}$. Na obrázku 2.3 je zobrazený sémantický strom pre S.

Poznamenávame, že pre každý vrchol v v sémantickom strome pre S je $I(v)$ podmnožina niektorej interpretácie pre S. Dôsledkom toho budeme $I(v)$ nazývať *čiastočnou interpretáciou* pre S.

Keď je Herbrandovská báza množiny S nekonečná, každý úplný sémantický strom pre S bude taktiež nekonečný. Ľahko vidieť, že úplný sémantický strom vyčerpávajúco preberá všetky možné interpretácie pre S. Ak S nie je splniteľná, tak S nemôže byť pravdivá na žiadnej z týchto interpretácií. Preto môžeme zastaviť rast stromu z vrchola v, ak $I(v)$ odmieta S. To nám umožňuje nasledujúcu definíciu.



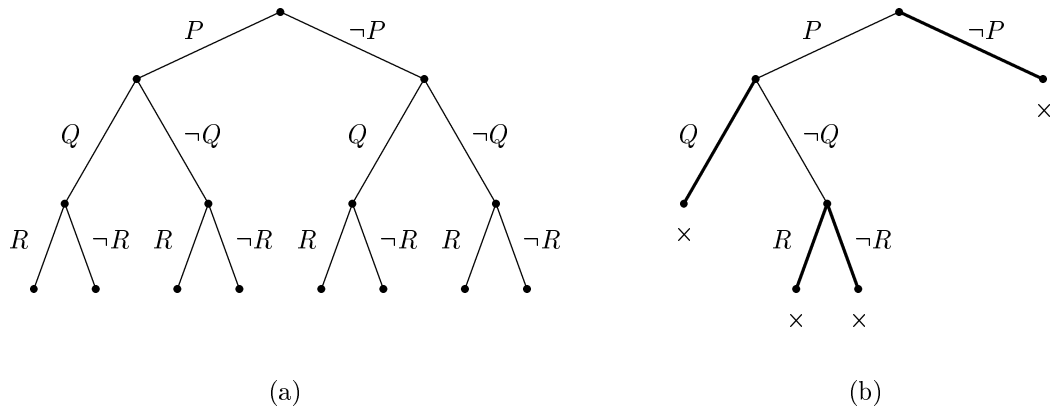
Obrázok 2.3. Strom k príkladu 2.24

Definícia 2.25. Vrchol v sémantického stromu pre S sa nazýva *odmietajúcim*, ak $I(v)$ odmieta niektorú základnú inštanciu klauzuly z S, no pre ľubovoľný predchádzajúci vrchol v' (v' predchádza v) $I(v')$ neodmieta žiadnu základnú inštanciu klauzuly z S.

Definícia 2.26. Hovoríme, že sémantický strom T je *uzavretý* práve vtedy, keď sa každá vetva vrchola T končí odmietajúcim vrcholom.

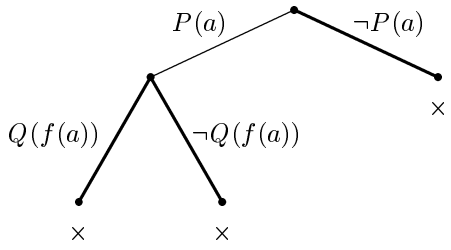
Definícia 2.27. Vrchol v uzavretého sémantického stromu nazývame *akceptujúcim*, ak všetky vrcholy bezprostredne nasledujúce za v sú odmietajúce.

Príklad 2.28. Nech $S = \{P, Q \vee R, \neg P \vee \neg Q, \neg R \vee \neg P\}$ Herbrandovská báza množiny S je $A = \{P, Q, R\}$. Na obrázku 2.4(a) je úplný sémantický strom pre S. Práve tak na obrázku 2.4(b) je uzavretý sémantický strom pre S.



Obrázok 2.4. Stromy k príkladu 2.28

Príklad 2.29. Uvažujme $S = \{P(x), P(x) \vee Q(f(x)), \neg Q(f(a))\}$. Herbrandovská báza množiny S je $A = \{P(a), Q(a), P(f(a)), Q(f(a)), \dots\}$. Na obrázku 2.5 je zobrazený uzavretý strom pre S.



Obrázok 2.5. Strom k príkladu 2.29

2.4. Herbrandova veta

Herbrandova veta je dôležitá veta v matematickej logike a tvorí základ väčšiny súčasných použiteľných algoritmov dokazovania teorém. Herbrandova veta úzko súvisí s vetou 2.4, t.j. aby sme preverili, či množina klauzúl nie je splniteľná, stačí nám uvažovať len interpretácie nad Herbrandovským univerzom S. Potom môžeme urobiť záver, že S nie je splniteľná. Zvyčajne to býva tak, že interpretácií je mnoho, možno aj nekonečne veľa. Nejakým spôsobom sa ich preto snažíme organizovať. Systematicky ich organizujeme pomocou sémantického stromu. Dokážeme dva varianty Herbrandovej vety. V literatúre sa takisto používajú oba varianty.

Veta 2.30 (Herbrandova). Množina klauzúl S nie je splniteľná práva vtedy, keď ľubovoľnému úplnému sémantickému stromu pre S zodpovedá konečný uzavretý sémantický strom, t.j. ľubovoľná vetva úplného stromu vedie do zamietajúceho vrchola.

Dôkaz: ▷ ($\Rightarrow$) Predpokladajme, že S nie je splniteľná. Nech T je úplný sémantický strom pre S. Pre každú vetvu V stromu T nech I_V je množina všetkých literálov, ktoré sú pripísané všetkým hranám vetvy V. Potom I_V je interpretácia pre S. Pretože S nie je splniteľná, musí I_V odmietať základnú inštanciu C' klauzuly C v S. Uvedomme si, že C' je konečná, teda na v musí existovať zamietajúci vrchol (ktorý má konečnú vzdialenosť od koreňa stromu). Pretože každá vetva stromu T má odmietajúci vrchol, existuje uzavretý sémantický strom T' pre S. Ďalej, pretože z každého vrchola v z T' vychádza len konečný počet hrán, tak T' musí byť konečný (t.j. počet vrcholov v T' je konečný), inak by sme v súlade s Königovou lemou mohli nájsť nekonečne dlhú vetvu, ktorá neobsahuje odmietajúce vrcholy. Tým je ukončený dôkaz prvej časti vety.

($\Leftarrow$) Obrátene, ak pre každý úplný sémantický strom T pre S existuje konečný uzavretý sémantický strom, tak každá vetva T obsahuje odmietajúci vrchol. To znamená, že každá interpretácia odmieta S. Teda S nie je splniteľná. To ukončuje dôkaz druhej časti vety. ◁

Veta 2.31 (Herbrandova). Množina klauzúl S nie je splniteľná práve vtedy, keď existuje konečná nesplniteľná množina S' základných inštancií klauzúl z S.

Dôkaz: ▷ ($\Rightarrow$) Predpokladajme, že S nie je splniteľná. Nech T je úplný sémantický strom pre S. Potom podľa Herbrandovej vety 2.30 existuje konečný uzavretý sémantický strom T' zodpovedajúci stromu T. Nech S' je množina všetkých základných inštancií klauzúl, ktoré sa odmietajú v zamietajúcich vrcholoch stromu T'. Množina S' je konečná, pretože v strome T' je len konečný počet zamietajúcich vrcholov. Pretože S' neplatí v žiadnej interpretácii pre S', tak S' nie je splniteľná.

($\Leftarrow$) Predpokladajme, že existuje konečná nesplniteľná množina S' základných inštancií klauzúl z S. To znamená, že pre každú interpretáciu I je S' odmietnutá (vyvrátená). Pretože každá interpretácia I pre S obsahuje interpretáciu I' množiny S' a I' zamieta S', tak I musí taktiež odmietať aj S, a teda S nie je splniteľná. ◁

Príklad 2.32. Nech $S = \{P(x), \neg P(f(a))\}$. Táto množina S nie je splniteľná. Z Herbrandovej vety vyplýva, že existuje konečná nesplniteľná množina S' základných inštancií klauzúl množiny S. Našli sme jednu z týchto množín: $S' = \{P(f(a)), \neg P(f(a))\}$.

Príklad 2.33. Nech $S = \{\neg P(x) \vee Q(f(x), x), P(g(b)), \neg Q(y, z)\}$. Množina S nie je splniteľná. Jedna z nesplniteľných množín základných inštancií klauzúl množiny S je $S' = \{\neg P\big(g(b)\big) \vee \vee\ Q\big(f(g(b)), g(b)\big), P\big(g(b)\big), \neg Q\big(f(g(b)), g(b)\big)\}$.

Príklad 2.34. Nech množina S obsahuje nasledujúce klauzuly:

$$\begin{aligned} S = \{&\neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(x, v, w) \vee P(u, z, w), \\ &\neg P(x, y, u) \vee \neg P(z, y, v) \vee \neg P(u, z, w) \vee P(x, v, w), \\ &P(g(x, y), x, y), P(x, h(x, y), y), P(x, y, f(x, y)), \neg P(k(x), x, k(x))\} \end{aligned}$$

Táto množina je tiež nesplniteľná, no nie je ľahké nájsť ihneď konečnú nesplniteľnú množinu S' základných inštancií klauzúl množiny S. Jedna cesta nájdenia takej množiny S' spočíva v zostrojení uzavretého sémantikcého stromu T' pre S. Potom množina S' všetkých základných inštancií odmietaných v zamietajúcich vrcholov stromu T' je hľadaná množina. Množinu S' uvádzame nižšie. Môžeme sa presvedčiť, že každá základná klauzula v S' je základná inštancia niektorej klauzuly z množiny S a že S' nie je splniteľná.

$$\begin{aligned} S' = \Big\{ &P\Big(a, h(a, a), a\Big), \neg P\Big(k\big(h(a, a)\big), h(a, a), k\big(h(a, a)\big)\Big), P\Big(g\big(a, k(h(a, a))\big), a, k\big(h(a, a)\big)\Big), \\ &\neg P\Big(g\big(a, k(h(a, a))\big), a, k\big(h(a, a)\big)\Big) \vee \neg P\Big(a, h(a, a), a\Big) \vee \\ &\vee \neg P\Big(g\big(a, k(h(a, a))\big), a, k\big(h(a, a)\big)\Big) \vee P\Big(k\big(h(a, a)\big), h(a, a), k\big(h(a, a)\big)\Big)\Big\} \end{aligned}$$

2.5. Aplikácie Herbrandovej vety

Druhý variant Herbrandovej vety predpokladá procedúru odmietnutia. To znamená, že ak treba dokázať nesplniteľnosť množiny klauzúl S a máme algoritmus pre počítač, ktorý dokáže úspešne indukovať množiny $S'_1, \ldots, S'_n$ základných inštancií klauzúl z S a úspešne stanoviť ich nesplniteľnosť, tak táto proceúra, ako nám garantuje Herbrandova veta, nám ukáže také konečné n, že S'_n nie je splniteľná.

Gilmore bol jeden z prvých, kto aplikoval uvedenú ideu (1960). Napísal program pre počítač, ktorý úspešne generoval množiny $S'_0, S'_1, \ldots$ zámenou premenných v S konštantami z H_i — množinami konštánt i-tej úrovne pre S. Pretože S'_i je konjunkcia základných inštancií, tak môžeme používať ľubovoľnú metódu vhodnú vo výrokovej logike, aby sme preverili jej nesplniteľnosť. T.j., uviedol každú indukovanú množinu S'_i v tvare d.n.f. Potom sa každá konjunkcia v d.n.f., obsahujúca kontrárne dvojice, vynecháva. Ak takýmto postupom získame prázdnu množinu pre niektoré S'_i, tak množina S_i je nesplniteľná.

Príklad 2.35. Uvažujme $S = \{P(x), \neg P(a)\}$, $H_0 = \{a\}$, $S'_0 = P(a) \wedge \neg P(a) = \square$. Tým je dokázané, že množina S nie je splniteľná.

Príklad 2.36. Nech $S = \{P(a), \neg P(x) \vee Q(f(x)), \neg Q(f(a))\}$, $H_0 = \{a\}$,

$$\begin{aligned} S'_0 &= P(a) \wedge \big(\neg P(a) \vee Q(f(a))\big) \wedge \neg Q(f(a)) = \\ &= \big(P(a) \wedge \neg P(a) \wedge \neg Q(f(a))\big) \vee \big(P(a) \wedge Q(f(a)) \wedge \neg Q(f(a))\big) = \square \vee \square = \square \end{aligned}$$

Tým je dokázané, že S nie je splniteľná.

Multiplikatívna metóda nie je efektívna. Možno sa ľahko presvedčiť, že pre množinu z desiatich dvojliterálnych základných klauzúl existuje 2^{10} konjunkcií. Davis a Putnam (1960) zaviedli efektívnejšiu metódu na preverovanie nesplniteľnosti množiny základných klauzúl. Opíšeme niektorú modifikáciu ich metódy.

2.6. Metóda Davisa a Putnama

Nech S je množina klauzúl. Podstatu metódy tvoria nasledujúce štyri pravidlá:

(1) ***Pravidlo tautológie:*** Vynecháme všetky tautologické základné inštancie klauzúl z S. Množina S', ktorá nám zostáva po vynechaní, nie je splniteľná práve vtedy, keď S nie je splniteľná.

(2) ***Pravidlo jednoliterálnych klauzúl:*** Ak existuje jednotková základná klauzula (rozmer klauzuly je rovný počtu literálov) L v S, tak S' dostávame z S vynechaním tých základných klauzúl v S, ktoré obsahujú L. Ak S' je prázdna množina, tak S je splniteľná. V opačnom prípade zostrojíme množinu S'', ak vynechávame z S' výskyty $\neg L$. S'' nie je splniteľná práve vtedy, keď S nie je splniteľná. poznamenávame, že ak $\neg L$ je jednotková základná klauzula, tak pri vyčiarkovaní $\neg L$ sa zmení na $\square$.

(3) ***Pravidlo čistých literálov:*** Literál L v základnej klauzule z S budeme nazývať *čistým* v S práve vtedy, ak sa literál $\neg L$ nevyskytuje v žiadnej základnej klauzule S. Ak je literál L čistý v S, tak vynecháme všetky základné klauzuly obsahujúce L. Množina S', ktorá zostala, nie je splniteľná práve vtedy, keď S nie je splniteľná.

(4) ***Pravidlo rezu:*** Ak množinu S môžeme vyjadriť v tvare

$$(A_1 \vee L) \wedge \cdots \wedge (A_m \vee L) \wedge (B_1 \vee \neg L) \wedge \cdots \wedge (B_n \vee \neg L) \wedge R$$

kde v A_i, B_i, a R sa nevyskytujú L ani $\neg L$, tak dostávame množiny (nazývame ich *množinami rezu*)

$$S_1 = A_1 \wedge \cdots \wedge A_m \wedge R$$
$$S_2 = B_1 \wedge \cdots \wedge B_n \wedge R$$

Pritom S nie je splniteľná práve vtedy, keď $S_1 \vee S_2$ nie je splniteľná, t.j. keď S_1 a S_2 nie sú splniteľné.

Môžeme teraz dokázať, že vyššie uvedené pravidlá môžeme aj obrátiť, t.j. ak požadovaná množina S nie je splniteľná, tak množina, ktorá zostáva po aplikácii jedného z pravidiel nie je splniteľná a obrátene.

Dôkaz pre pravidlo (1): $\triangleright$ Pretože tautológia vyhovuje každej interpretácii, S' nie je splniteľná práve vtedy, keď S nie je splniteľná. $\triangleleft$

Dôkaz pre pravidlo (2): $\triangleright$ Ak S' je prázdna množina, tak všetky základné klauzuly z S obsahujú L, a teda každá interpretácia obsahujúca L vyhovuje S. Preto je S splniteľná. Musíme ešte dokázať, že S'' nie je splniteľná práve vtedy, keď S nie je splniteľná.

Predpokladajme, že S'' nie je splniteľná. Ak je splniteľná S, tak existuje model M, ktorý obsahuje L. Ďalej, pretože M odmieta $\neg L$, modelu M musia vyhovovať všetky klauzuly, ktoré spočiatku obsahovali $\neg L$. Z toho vyplýva, že M musí vyhovovať S''. To však protirečí predpokladu, že S'' nie je splniteľná. Preto S nie je splniteľná.

Obrátene, predpokladáme, že S nie je splniteľná. Ak S'' splniteľná je, tak existuje model M'' pre S''. Takým spôsobom každá interpretácia S obsahujúca M'' aj L musí byť modelom pre S. To je v spore s predpokladom, že S nemá model. Preto S'' musí byť nesplniteľná. Z toho vyplýva, že S'' nie je splniteľná práve vtedy, keď S nie je splniteľná. $\triangleleft$

Dôkaz pre pravidlo (3): $\triangleright$ Predpokladajme, že S' nie je splniteľná. Potom S nemôže byť splniteľná, pretože S' je podmnožinou S. Obrátene, predpokladajme, že S nie je splniteľná. Ak S' splniteľná je, tak existuje model M pre S', pričom ani L, ani $\neg L$ sa nenachádzajú v M. Takým spôsobom každá interpretácia S, ktorá obsahuje M aj L je model S. To je v spore s predpokladom, že S nemá model. Preto S' nemôže byť splniteľná. Z toho vyplýva, že S' nie je splniteľná práve vtedy, keď S nie je splniteľná. $\triangleleft$

Dôkaz pre pravidlo (4): $\triangleright$ Predpokladajme, že S nie je splniteľná. Ak $(S_1 \vee S_2)$ je splniteľná, tak buď S_1 alebo S_2 má model. Ak S_1 (resp. S_2) má model M, tak každá interpretácia S obsahujúca M a $\neg L$ (resp. L) je model pre S. To je v spore s predpokladom, že S nemá model. Z toho vyplýva, že $(S_1 \vee S_2)$ nie je splniteľná.

Predpokladajme, že $(S_1 \vee S_2)$ nie je splniteľná. Ak je S splniteľná, tak S musí mať model M. Ak M obsahuje $\neg L$ (resp. L), tak M vyhovuje S_1 (resp. S_2). To je v spore s predpokladom, že $(S_1 \vee S_2)$ nie je splniteľná. Preto S nemôže byť splniteľná. Z toho vyplýva, že S nie je splniteľná práve vtedy, keď nie je splniteľná $(S_1 \vee S_2)$. $\lhd$

Uvedené pravidlá sú dôležité. V nasledujúcich častiach uvidíme, že tieto pravidlá majú širokú oblasť aplikovateľnosti. Teraz uvedieme niekoľko príkladov, aby sme demonštrovali používanie týchto pravidiel.

Príklad 2.37. Ukážte, že $S = (P \vee Q \vee \neg R) \wedge (P \vee \neg Q) \wedge \neg P \wedge R \wedge U$ nie je splniteľná.

Riešenie:

(1) $(P \vee Q \vee \neg R) \wedge (P \vee \neg Q) \wedge \neg P \wedge R \wedge U$
(2) $(Q \vee \neg R) \wedge (\neg Q) \wedge R \wedge U$ — pravidlo (2) v $\neg P$
(3) $\neg R \wedge R \wedge U$ — pravidlo (2) v $\neg Q$
(4) $\square \wedge U$ — pravidlo (2) v $\neg R$

Teda vidíme, že posledná formula obsahuje prázdnu klauzulu $\square$, a preto S nie je splniteľná.

Príklad 2.38. Ukážte, že $S = (P \vee Q) \wedge \neg Q \wedge (\neg P \vee Q \vee \neg R)$ je splniteľná.

Riešenie:

(1) $(P \vee Q) \wedge \neg Q \wedge (\neg P \vee Q \vee R)$
(2) $P \wedge (\neg P \vee \neg R)$ — pravidlo (2) v $\neg Q$
(3) $\neg R$ — pravidlo (2) v P
(4) ■ — pravidlo (2) v $\neg R$

t.j. zvolíme $I = \{P, \neg Q, \neg R\}$. Posledná množina je prázdna množina. Z toho vyplýva, že S je splniteľná pre $I = \{P, \neg Q, \neg R\}$.

Príklad 2.39. Ukážte, že množina $S = (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (Q \vee \neg R) \wedge (\neg Q \vee \neg R)$ je splniteľná.

Riešenie:

(1) $(P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (Q \vee \neg R) \wedge (\neg Q \vee \neg R)$
(2) $\big(\neg Q \wedge (Q \vee \neg R) \wedge (\neg Q \vee \neg R)\big) \vee$
$\vee \big(Q \wedge (Q \vee \neg R) \wedge (\neg Q \vee \neg R)\big)$ — pravidlo (2) v P
(3) $\neg R \vee \neg R$ — pravidlo (2) v $\neg Q$ a Q
(4) ■ $\vee$ ■ — pravidlo (2) v $\neg R$

Pretože obidve množiny S_1 aj S_2 sú splniteľné, je aj S splniteľná.

Príklad 2.40. Ukážte, že $S = (P \vee Q) \wedge (P \vee \neg Q) \wedge (R \vee Q) \wedge (R \vee \neg Q)$ je splniteľná.

Riešenie:

(1) $(P \vee Q) \wedge (P \vee \neg Q) \wedge (R \vee Q) \wedge (R \vee \neg Q)$
(2) $(R \vee Q) \wedge (R \vee \neg Q)$ — pravidlo (3) v P
(3) ■ — pravidlo (3) v R

Teda, S je splniteľná.

Uvedená metóda na preverovanie nesplniteľnosti je efektívnejšia ako multiplikatívna metóda a môže byť aplikovaná na ľubovoľnú formulu vo výrokovej logike, t.j. najprv vyjadríme danú logickú formulu v konjunktívnej normálnej forme a potom aplikujeme vyššie uvedené štyri pravidlá.

Ľahko sa môžeme presvedčiť, že pravidlá (2) a (3) sú špeciálny prípad pravidla (4). Je prirodzené predpokladať, že pri aplikácii pravidla (4) podľa literálu L klauzuly obsahujúce $L \vee \neg L$ vynecháme. Takýmto spôsobom je pravidlo (1) zahrnuté do pravidla (4). Pravidlá (2) a (3) zodpovedajú tvorbe jednej z množín S_1 alebo S_2. Pravidlá (1)–(4) sa aplikujú na základné klauzuly.

KAPITOLA 3

Rezolvenčná metóda

Z Herbrandovej vety (II. variant) vyplýva jednoduchá metóda pre nájdenie zamietnutia. Táto jednoduchá metóda postupného preberania má jeden podstatný nedostatok: musíme generovať množiny $S'_0, S'_1, S'_2, \ldots$ základných inštancií klauzúl. Vo väčšine prípadov táto postupnosť rastie exponenciálne.

Základná idea rezolvenčnej metódy spočíva v zistení, či S obsahuje prázdnu klauzulu $\square$. Ak S obsahuje $\square$, tak S nie je splniteľná. Ak S neobsahuje $\square$, tak preverujeme nesledujúci fakt: môžeme $\square$ získať z S. Neskôr uvidíme, že na základe Herbrandovej vety (I. variant) preverenie získania $\square$ je ekvivalentné spočítaniu počtu vrcholov uzavretého sématického stromu pre S.

Podľa uvedeného variantu Herbrandovej vety S nie je splniteľná práve vtedy, keď existuje konečný uzavretý sémantický strom T pre S. Je zrejmé, že S obsahuje $\square$ práve vtedy, keď sa T skladá len z jedného vrchola — koreňa. Ak S neobsahuje $\square$, tak T nemusí obsahovať viacej ako jeden vrchol. No ak môžeme zostrojiť strom T s jedným vrcholom, tak sa nakoniec $\square$ nutne objaví v S. V tom spočíva podstata metódy rezolvent. Inými slovami povedané: môžeme ju chápať ako špeciálne pravidlo odvodenia, ktoré použijeme na tvorbu nových klauzúl z S. Ak pridáme tieto nové klauzuly k S, tak niektoré vrcholy v počiatočnom T sa stávajú odmietajúcimi vrcholmi. Takýmto spôsobom môže byť počet vrcholov v T zmenšený a nakoniec prázdnu klauzulu $\square$ môžeme získať.

Najprv budeme uvažovať metódu rezolvent pre výrokovú logiku. Potom ju rozšírime na logiku 1. rádu.

3.1. Metóda rezolvent pre výrokovú logiku

Najprv sformulujeme *pravidlo rezolventy*; niekedy ho budeme nazývať aj *pravidlo rezu*:

Definícia 3.1. Nech C_1 a C_2 sú ľubovoľné dve klauzuly. Ak existuje literál L_1 v C_1, ktorý je kontrárny literálu L_2 v C_2, tak vynechávame L_1 a L_2 z C_1, resp. C_2 a zostrojíme disjunkciu zostávajúcich klauzúl. Klauzulu, ktorá vznikne takýmto spôsobom nazývame *rezolventa* C_1 a C_2.

Príklad 3.2. Uvažujme nasledujúce klauzuly

$$C_1\colon P \vee R \qquad\qquad C_2\colon \neg P \vee Q$$

Klauzula C_1 má literál P, ktorý je kontrárny k literálu $\neg P$ v C_2. Ak teda vynecháme P a $\neg P$ z C_1, resp. C_2 a utvoríme disjunkciu zostávajúcich klauzúl R a Q, dostávame rezolventu $R \vee Q$.

Príklad 3.3. Uvažujme klauzuly

$$C_1\colon \neg P \vee Q \vee R \qquad\qquad C_2\colon \neg Q \vee S$$

Rezolventa C_1 a C_2 je $\neg P \vee R \vee S$.

Príklad 3.4. Uvažujme klauzuly

$$C_1\colon \neg P \vee Q \qquad\qquad C_2\colon \neg P \vee R$$

Pretože neexistuje žiadny literál v C_1, ktorý je kontrárny nejakému literálu v C_2, tak neexistuje žiadna rezolventa C_1 a C_2.

Dôležitou vlastnosťou rezolventy je to, že ľubovoľná rezolventa dvoch klauzúl C_1 a C_2 je logický dôsledok C_1 a C_2. Túto vlastnosť dokážeme v nasledujúcej vete.

Veta 3.5. Nech sú dané dve klauzuly C_1 a C_2. Potom rezolventa C klauzúl C_1 a C_2 je logickým dôsledkom C_1 a C_2.

Dôkaz: ▷ Nech C_1, C_2 a C majú nasledujúci význam:

$$C_1 = L \vee C_1'$$
$$C_2 = \neg L \vee C_2'$$
$$C = C_1' \vee C_2'$$

kde C_1' a C_2' sú disjunkcie literálov. Predpokladajme, že C_1 a C_2 sú pravdivé v interpretáci I. Chceme ukázať, že rezolventa C klauzúl C_1 a C_2 je taktiež pravdivá v I. Predpkladajme, že L nie je pravdivý v I. Potom C_1 nemôže byť jednotková klauzula, inak by C_1 bola nepravdivá v I. Analogicky môžeme dokázať, že ak $\neg L$ neplatí v I, tak C_2' musí byť pravdivá v I, čo bolo treba dokázať. ◁

Poznámka 3.6. Ak máme dve jednotkové klauzuly, tak ich rezolventa, ak existuje, je prázdna klauzula □. To nás privádza k záveru, že pre nesplniteľnú množinu klauzúl aplikáciou pravidla rezolvent môžeme dostať □. Tento výsledok dokážeme neskôr. Zatiaľ uvedieme definíciu odvodenia.

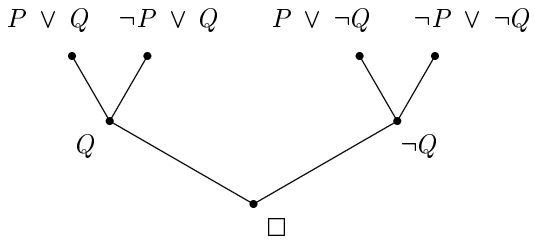
Definícia 3.7. Nech S je množina klauzúl. *Rezolvenčným odvodením* C z S je taká konečná postupnosť $C_1, C_2, \ldots, C_n$ klauzúl, že každá C_k patrí do S alebo je rezolventou predchádzajúcich C_i a C_j a že $C_n = C$. Odvodenie □ z S nazývame *zamietnutie* (alebo *dôkaz nesplniteľnosti*) S. Hovoríme, že klauzulu C môžeme *odvodiť* alebo *získať* z S, ak existuje odvodenie C z S.

Uvedieme niekoľko príkladov, ktoré ilustrujú použitie metódy rezolvent pre dôkaz nesplniteľnosti množiny klauzúl.

Príklad 3.8. Uvažujme množinu $S = \{(1)\ \neg P \vee Q, (2)\ \neg Q, (3)\ P\}$. Z (1) a (2) dostávame rezolventu (4) $\neg P$. Zo (4) a (3) dostávame □. Pretože □ dostávame z S aplikovaním pravidla rezolventy, tak v súlade s predchádzajúcou vetou, □ je logický dôsledok S. Z toho vyplýva, že S nie je splniteľná.

Príklad 3.9. Pre množinu $S = \{(1)\ P \vee Q, (2)\ \neg P \vee Q, (3)\ P \vee \neg Q, (4)\ \neg P \vee \neg Q\}$ dostávame nasledujúce rezolventy: (5) Q z (1) a (2), (6) $\neg Q$ z (3) a (4), (7) □ z (5) a (6).

Dostali sme □, teda S nie je splniteľná. Uvedené odvodenie môžeme vyjadriť pomocou stromu, ktorý nazývame *strom odvodenia*.



Obrázok 3.1

Pravidlo rezolventy je veľmi silné odvodzovacie pravidlo. V ďalšom ho budeme definovať aj pre logiku 1. rádu. Taktiež dokážeme úplnosť metódy rezolvent pre dôkaz nesplniteľnosti množiny klauzúl, t.j. daná množina klauzúl nie je splniteľná práve vtedy, keď existuje odvodenie prázdnej klauzuly □ z S. Neskôr uvedieme aj príklady aplikovania metódy rezolvent. Na záver tejto časti odvodíme ekvivalentnosť pravidla rezu a pravidla modus ponens. To značí nasledujúci fakt:

(a) $\neg A \vee B, A \vee C \vdash B \vee C$ — Ak prepíšeme uvedené tvrdenie pomocou implikácií dostávame $A \to B, \neg A \to C \vdash \neg B \to C$. Ukážeme, že použitím pravidla modus ponens dokážeme uvedené tvrdenie:

(1) $\vdash (A \to B) \to (\neg B \to \neg A)$
(2) $\vdash (\neg B \to \neg A) \to ((\neg A \to C) \to (\neg B \to C))$

(3) $\vdash \neg B \to C$, z predpokladov $A \to B$, $\neg A \to C$ použitím pravidla modus ponens (2-krát).
(b) $A, A \to B \vdash B$ — Z uvedených predpokladov pomocou pravidla rezu odvodíme B: najprv prepíšeme predpoklady pomocou disjunkcie; dostávame $A \vee \square, \neg A \vee B \vdash B \vee \square = B$.

3.2. Substitúcia a unifikácia

V predchádzajúcej časti sme metódu rezolvent uvažovali pre výrokovú logiku. V ďalších častiach sa budeme snažiť rozšíriť túto metódu na logiku 1. rádu. V predchádzajúcej časti sme poznamenali, že podstatné pre pravidlo rezu je nájsť v klauzule literál, ktorý je kontrárny literálu v druhej klauzule. Pre klauzuly, ktoré neobsahujú premenné, je to veľmi jednoduché. No pre klauzuly obsahujúce premenné, je to zložitejšia vec. Uvažujme napr. klauzuly

$$C_1 = P(x) \vee Q(x) \qquad C_2 = \neg P(f(x)) \vee R(x)$$

Neexistuje žiaden literál v C_1, kontrárny nejakému literálu v C_2. No ak zameníme premennú x v C_1 na $f(a)$ a x v C_2 na a, tak dostávame:

$$C_1' = P(f(a)) \vee Q(f(a)) \qquad C_2 = \neg P(f(a)) \vee R(a)$$

Vieme, že C_1' a C_2' sú základné inštancie C_1, resp. C_2 a $P(f(a))$ a $\neg P(f(a))$ sú kontrárne navzájom. Z toho vyplýva, že z C_1' a C_2' môžeme dostať rezolventu $C_3' = Q(f(a)) \vee R(a)$.

Vo všeobecnom prípade, ak zameníme x v C_1 funkciou $f(x)$, dostaneme $C_1^* = P(f(x)) \vee \vee Q(f(x))$. Opäť C_1^* je inštancia C_1. Súčasne literál $P(f(x))$ v C_1^* je kontrárny literálu $\neg P(f(x))$ v C_2. Z toho vyplýva, že môžeme dostať rezolventu z C_1^* a C_2: $C_3 = Q(f(x)) \vee R(x)$, C_3' je inštancia klauzuly C_3. Ak vhodnými termami zamieňame premenné v C_1 a C_2, ako to robíme vyššie, tak môžeme dostať nové rezolventy C_1 a C_2. Okrem toho klauzula C_3 je najviac spoločnou klauzulou v tom zmysle, že všetky druhé klauzuly, ktoré dostaneme podobne ako vyššie, sú inštancie C_3. C_3 taktiež nazveme *rezolventou* C_1 a C_2. Ďalej sa budemem zaoberať tým, ako tvoriť rezolventy z klauzúl (obsahujúcich aj premenné). Pretože získanie rezolvent z klauzúl často potrebuje zamieňať premenné, uvedieme potrebné definície.

Definícia 3.10. Pod *substitúciou* rozumieme konečnú množinu tvaru $\{t_1/v_1, \ldots, t_n/v_n\}$, kde každá v_i je premenná, každý term t_i je rôzny od v_i a všetky v_i sú navzájom rôzne. Ak $t_1, t_2, \ldots t_n$ sú základné termy, tak substitúciu nazývame *základná substitúcia*. Substitúciu, ktorá neobsahuje žiaden prvok nazývame *prázdna* a označujeme ju ε. Na označenie substitúcií budeme používať grécke písmená.

Príklad 3.11. Nasledujúce dve množiny sú substitúcie:

$$\{f(z)/x, y/z\} \qquad \{a/x, g(y)/y, f(g(b))/z\}$$

Definícia 3.12. Nech $\theta = \{t_1/v_1, \ldots t_n/v_n\}$ je substitúcia a E je výraz. Potom $E\theta$ je výraz, ktorý dostaneme z E tak, že súčasne zameníme všetky výskyty premennej v_i $(1 \leq i \leq n)$ v E termom t_i. $E\theta$ nazývame *inštancia* E.

Príklad 3.13. Nech $\theta = \{a/x, f(b)/y, c/z\}$ a $E = P(x, y, z)$. Potom $E\theta = P(a, f(b), c)$.

Definícia 3.14. Nech $\theta = \{t_1/x_1, \ldots, t_n/x_n\}$ a $\lambda = \{u_1/y_1, \ldots, u_m/y_m\}$ sú dve substitúcie. Potom *zloženie (kompozícia)* θ a λ je substitúcia (označíme je $\theta \circ \lambda$), ktorú dostávame z množiny $\{t_1\lambda/x_1, \ldots, t_n\lambda/x_n, u_1/y_1, \ldots, u_m/y_m\}$ vynechaním všetkých prvkov $t_j\lambda/x_j$, pre ktoré $t_j\lambda = x_j$ a všetkých prvkov u_i/y_i takých, že $y_i \in \{x_1, x_2, \ldots, x_n\}$.

Príklad 3.15. Nech $\theta = \{t_1/x_1, t_2/x_2\} = \{f(y)/x, z/y\}$, $\lambda = \{u_1/y_1, u_2/y_2, u_3/y_3\} = \{a/x, b/y, y/z\}$. Potom $\{t_1\lambda/x_1, t_2\lambda/x_2, u_1/y_1, u_2/y_2, u_3/y_3\} = \{f(b)/x, y/y, a/x, b/y, y/z\}$. Pretože $t_2\lambda = x_2$, $t_2\lambda/x_2$ (t.j. y/y) musí byť vynechané z množiny. Ďalej y_1 a y_2 sú obsiahnuté v $\{x_1, x_2\}$, teda u_1/y_1 a u_2/y_2 (t.j. a/x a b/y) musíme vynechať. Takým spôsobom dostávame $\theta \circ \lambda = \{f(b)/x, y/z\}$.

Poznamenávame, že kompozícia zámen je asociatívna a že prázdna zámena ε je súčasne ľavá aj pravá identita, t.j. $(\theta \circ \lambda) \circ \mu = \theta \circ (\lambda \circ \mu)$ a $\varepsilon \circ \theta = \theta \circ \varepsilon$ pre všetky θ, λ a μ (zámeny, substitúcie tvoria monoid, t.j. pologrupu s 1).

Pri dôkazoch metódou rezolvent, aby sme mohli identifikovať kontrárne dvojice literálov, je často treba zjednotiť — *unifikovať* — dva alebo viacej výrazov, t.j. musíme nájsť zámenu, ktorá môže previesť niekoľko výrazov na identické. V ďalšom sa budeme zaoberať unifikáciou výrazov.

Definícia 3.16. Substitúciu θ nazývame *unifikátorom* pre množinu $\{E_1, E_2, \ldots, E_k\}$ práve vtedy, keď $E_1\theta = E_2\theta = \cdots = E_k\theta$. Hovoríme, že množina $\{E_1, E_2, \ldots, E_k\}$ je *unifikovateľná*, ak pre ňu existuje unifikátor.

Definícia 3.17. Unifikátor σ pre množinu výrazov nazývame *najvšeobecnejším* unifikátorom práve vtedy, ak pre každý unifikátor θ pre túto množinu existuje taká substitúcia λ, že $\theta = \sigma \circ \lambda$.

Príklad 3.18. Množina $\{P(a, y), P(x, f(b))\}$ je unifikovateľná, pretože substitúcia $\theta = \{a/x, f(b)/y\}$ je jej unifikátor.

3.3. Unifikačný algoritmus

V tejto časti uvedieme algoritmus unifikácie pre nájdenie najvšeobecnejšieho unifikátora pre konečnú neprázdnu unifikovateľnú množinu výrazov. Keď množina nie je unifikovateľná, algoritmus zaznamená aj tento fakt.

Uvažujme $P(a)$ a $P(x)$. Tieto dva výrazy nie sú identické. Diferencia je v tom, že a sa vyskytuje v $P(a)$ a x v $P(x)$. Aby sme mohli $P(a)$ a $P(x)$ stotižniť, najprv musíme nájsť diferenciu a potom sa pokúsiť túto diferenciu vylúčiť. Pre $P(a)$ a $P(x)$ diferencia bude $\{a, x\}$. Pretože x je premenná, tak x môžeme zameniť na a. Na tom je založená idea unifikačného algoritmu.

Definícia 3.19. *Diferenčnú množinu* neprázdnej množiny výrazov W dostávame tak, že nájdeme prvú (zľava) pozíciu, na ktorej sa nie pre všetky výrazy z W nachádza jeden a ten istý symbol a vypíšeme z každého výrazu v W podvýrazy, ktoré sa začínajú symbolom, ktorý sa nachádza na uvažovanej pozícii. Množina D týchto podvýrazov sa nazýva *diferenčná množina* pre W a jej výrazy sú termy.

Príklad 3.20. Ak $W = \{P(x, \underline{f(y, z)}), P(x, \underline{a}), P(x, \underline{g(h(k(x)))})\}$, tak prvá pozícia, na ktorej sa nie vo všetkých výrazoch z W nachádzajú rovnaké symboly, je piata, pretože všetky výrazy majú rovnaké prvé 4 symboly, a to „$P(x,$". Takým spôsobom sa diferenčná množina skladá zo zodpovedajúcich výrazov (podvýrazov) — podčiarknutých termov, ktoré sa začínajú na piatej pozícii, teda je to množina $\{f(y, z), a, g(h(k(x)))\}$.

Unifikačný algoritmus

Krok 1. $k = 0$, $W_0 = W$, $\sigma_0 = \varepsilon$.

Krok 2. Ak W_k je jednotková klauzula, algoritmus zakončí svoju činnosť: σ_k je najvšeobecnejší unifikátor pre W. V opačnom prípade nájdeme D_k — diferenčnú množinu pre W_k.

Krok 3. Ak existujú také elementy v_k a t_k v D_k, že v_k je premenná, ktorá sa nevyskytuje v t_k, tak prejdeme ku kroku 4. V opačnom prípade algoritmus zakončuje svoju činnosť: W nie je unifikovateľná.

Krok 4. Nech $W_{k+1} = W_k\{t_k/v_k\}$ a $\sigma_{k+1} = \sigma_k \circ \{t_k/v_k\}$ (poznamenávame, že $W_{k+1} = W_k\sigma_{k+1}$).

Krok 5. Vypíšeme hodnoty pre $k + 1$ a prejdeme ku kroku 2.

Obrázok 3.2. Unifikačný algoritmus

Príklad 3.21. Nájdite najvšeobecnejší unifikátor pre $W = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}$.

(1) $\sigma_0 = \varepsilon$ a $W_0 = W$. Pretože W_0 nie je jednotková klauzula, tak σ_0 nie je najvšeobecnejší unifikátor pre W.

(2) Diferenčná množina $D_0 = \{a, z\}$. V D_0 existuje premenná $v_0 = z$, ktorá sa nevyskytuje v $t_0 = a$.

(3) Nech

$$\sigma_1 = \sigma_0 \circ \{t_0/v_0\} = \varepsilon \circ \{a/z\} = \{a/z\}$$
$$W_1 = W_0\{t_0/v_0\} = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}\{a/z\} =$$
$$= \{P(a, x, f(g(y))), P(a, f(a), f(u))\}$$

(4) W_1 nie je jednotková klauzula, našli sme diferenčnú množinu D_1 pre W_1, a to $D_1 = \{x, f(a)\}$.

(5) Z D_1 dostávame, že $v_1 = x$ a $t_1 = f(a)$.

(6) Nech

$$\sigma_2 = \sigma_1 \circ \{t_1/v_1\} = \{a/z\} \circ \{f(a)/x\} = \{a/z, f(a)/x\}$$
$$W_2 = W_1\{t_1/v_1\} = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}\{f(a)/x\} =$$
$$= \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$$

(7) W_2 nie je jednotková klauzula, pretože sme našli diferenčnú množinu D_2 pre W_2, a to $D_2 = \{g(y), u\}$. Z D_2 dostávame, že $v_2 = u$ a $t_2 = g(y)$.

(8) Nech

$$\sigma_3 = \sigma_2 \circ \{t_2/v_2\} = \{a/z, f(a)/x\} \circ \{g(y)/u\} = \{a/z, f(a)/x, g(y)/u\}$$
$$W_3 = W_2\{t_2/v_2\} = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}\{g(y)/u\} =$$
$$= \{P(a, f(a), f(g(y))), P(a, f(a), f(g(y)))\} = \{P(a, f(a), f(g(y)))\}$$

(9) Pretože W_3 je jednoprvková klauzula, tak $\sigma_3 = \{a/z, f(a)/x, g(y)/u\}$ je najvšeobecnejší unifikátor pre W.

Príklad 3.22. Zistite, či je unifikovateľná množina $W = \{Q(f(a), g(x)), Q(y, y)\}$.

(1) Nech $\sigma_0 = \varepsilon$ a $W_0 = W$.

(2) W_0 nie je jednotková klauzula, pretože sme našli diferenčnú množinu D_0 pre W_0, a to $D_0 = \{f(a)/y\}$. Z D_0 vieme, že $v_0 = y$ a $t_0 = f(a)$.

(3) Nech

$$\sigma_1 = \sigma_0 \circ \{t_0/v_0\} = \varepsilon \circ \{f(a)/y\} = \{f(a)/y\}$$
$$W_1 = W_0\{t_0/v_0\} = \{Q(f(a), g(x)), Q(f(a), f(a))\}$$

(4) W_1 nie je jednotková klauzula, pretože nájdeme diferenčnú množinu D_1 pre W_1, a to $D_1 = \{g(x), f(a)\}$, a naviac nemáme prvok, ktorý by bol premennou. Teda unifikačný algoritmus končí svoju činnosť; môžeme urobiť záver, že W nie je unifikovateľná množina.

Poznamenávame, že vyššie uvedený algoritmus unifikácie vždy zakončuje svoju činnosť pre ľubovoľnú konečnú neprázdnu množinu výrazov, v opačnom prípade by vznikla nekonečná postupnosť $W\sigma_0, W\sigma_1, W\sigma_2 \ldots$ konečných neprázdnych množín, ktorá má tú vlastnosť, že každá nasledujúca množina má o jednu premennú menej ako predchádzajúca (skutočne: $W\sigma_k$ obsahuje v_k, no $W\sigma_{k+1}$ ju neobsahuje). No to nie je možné, pretože W obsahuje len konečný počet premenných.

Na príklade sme ukázali, že pre unifikovateľnú množinu W unifikačný algoritmus nájde najvšeobecnejší unifikátor. Že to ide urobiť vždy, dokazuje nasledujúca veta.

Veta 3.23 (Unifikačná veta). Ak W je konečná neprázdna unifikovateľná množina výrazov, tak unifikačný algoritmus vždy zakončuje svoju činnosť na 2. kroku a posledné σ_k bude najvšeobecnejší unifikátor pre W.

Dôkaz: ▷ Pretože W je unifikovateľná množina, tak θ označme jej ľubovoľný unifikátor. Indukciou vzhľadom na k ukážeme, že existuje taká substitúcia λ_k, že $\theta = \sigma_k \circ \lambda_k$.

1° Nech $k = 0$. Položme $\lambda_0 = \theta$. Potom $\theta = \sigma_0 \circ \lambda_0$, pretože $\sigma_0 = \varepsilon$.

2° Predpokladajme teraz, že $\theta = \sigma_k \circ \lambda_k$ platí pre $0 \leq k \leq n$. Ak $W\sigma_n$ je jednotková formula, tak algoritmus unifikácie zakončuje svoju činnosť na 2. kroku. Pretože $\theta = \sigma_n \circ \lambda_n$, tak σ_n bude najvšeobecnejší unifikátor pre W. Ak $W\sigma_n$ nie je jednotková klauzula, tak unifikačný algortimus nájde diferenčnú množinu D_n pre $W\sigma_n$. Pretože $\theta = \sigma_n \circ \lambda_n$ je unifikátor pre W, tak λ_n musí unifikovať D_n. Pretože D_n je diferenčná množina, tak v D_n musí existovať premenná v_n.

Nech t_n je ľubovoľný iný element rôzny od v_n. Pretože λ_n unifikuje D_n, tak $v_n\lambda_n = t_n\lambda_n$. Ak sa v_n vyskytuje v t_n, tak sa $v_n\lambda_n$ vyskytuje v $t_n\lambda_n$. No to nie je možné, pretože v_n a t_n sú rôzne a $v_n\lambda_n = t_n\lambda_n$. Z toho vyplýva, že v_n sa nevyskytuje v t_n. Preto sa unifikačný algoritmus nezastaví na 3. kroku, ale prejde ku 4. kroku k množine $W\sigma_{n+1}$, kde $\sigma_{n+1} = \sigma_n \circ \{t_n/v_n\}$.

Nech $\lambda_{n+1} = \lambda_n - \{t_n\lambda_n/v_n\}$. Pretože v_n sa nevyskytuje v t_n, tak

$$t_n\lambda_{n+1} = t_n(\lambda_n - \{t_n\lambda_n/v_n\}) = t_n\lambda_n$$

Takým spôsobom dostávame

$$\begin{aligned}\{t_n/v_n\} \circ \lambda_{n+1} = \{t_n\lambda_{n+1}/v_n\} \cup \lambda_{n+1} = \{t_n\lambda_n/v_n\} \cup \lambda_{n+1} = \\ = \{t_n\lambda_n/v_n\} \cup (\lambda_n - \{t_n\lambda_n/v_n\}) = \lambda_n\end{aligned}$$

To znamená, že $\lambda_n = \{t_n/v_n\} \circ \lambda_{n+1}$. Z toho vyplýva, že

$$\theta = \sigma_n \circ \lambda_n = \sigma_n \circ \{t_n/v_n\} \circ \lambda_{n+1} = \sigma_{n+1} \circ \lambda_{n+1}$$

Preto pre všetky $k \geq 0$ existuje taká substitúcia λ_k, že $\theta = \sigma_k \circ \lambda_k$.

Pretože unifikačný algoritmus musí skončiť svoju činnosť a neskončil ju na 3. kroku, tak musí svoju činnosť skončiť na 2. kroku. Okrem toho, pretože $\theta = \sigma_k \circ \lambda_k$ pre všetky k, tak posledná σ_k bude najvšeobecnejším unifikátorom pre W, čo sme potrebovali dokázať. ◁

3.4. Metóda rezolvent pre logiku prvého rádu

Po uvedení unifikačného algoritmu môžeme rozobrať metódu rezolvent pre logiku prvého rádu.

Definícia 3.24. Nech C je klauzula. Ak dva alebo viacej literálov (s rovnakým znakom predikátu) klauzuly C majú najvšeobecnejší unifikátor σ, tak $C\sigma$ sa nazýva *spojením* C. Ak $C\sigma$ je jednotková formula, tak sa toto spojenie nazýva *jednotkovým* spojením.

Príklad 3.25. Nech $C = \{\underline{P(x)} \vee \underline{P(f(y))} \vee \neg Q(x)$. Potom prvý a druhý podčiarknutý literál majú najvšeobecnejší unifikátor $\sigma = \{f(y)/x\}$. Z toho vyplýva, že $C\sigma = P(f(y)) \vee \neg Q(f(y))$ je spojenie C.

Definícia 3.26. Nech C_1 a C_2 sú dve klauzuly (nazývame ich *predpoklady*), ktoré nemajú žiadne spoločné premenné. Nech $L_1 \in C_1$ a $L_2 \in C_2$ sú dva literály. Ak L_1 a $\neg L_2$ majú najvšeobecnejší unifikátor σ, tak sa klauzula

$$(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$$

nazýva *(binárnou) rezolventou* C_1 a C_2. Literály L_1 a L_2 sa nazývajú *nadbytočné* a môžeme ich vynechať.

Príklad 3.27. Nech $C_1 = P(x) \vee Q(x)$ a $C_2 = \neg P(a) \vee R(x)$. Pretože x vystupuje v C_1 a C_2, tak zameníme premennú v C_2, teda nech $C_2 = \neg P(a) \vee R(y)$. Vyberme $L_1 = P(x)$ a $L_2 = \neg P(a)$. Pretože $\neg L_2 = P(a)$, tak L_1 a $\neg L_2$ majú najvšeobecnejší unifikátor $\sigma = \{a/x\}$. Z toho vyplýva, že

$$\begin{aligned}(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma) = (\{P(a), Q(a)\} - \{P(a)\}) \cup (\{\neg P(a), R(y)\} - \{\neg P(a)\}) = \\ = \{Q(a)\} \cup \{R(y)\} = \{Q(a), R(y)\} = Q(a) \vee R(y)\end{aligned}$$

Takýmto spôsobom $Q(a) \vee R(y)$ je binárna rezolventa C_1 a C_2. $P(x)$ a $\neg P(a)$ sú nadbytočné literály.

Definícia 3.28. *Rezolventou* predpokladov C_1 a C_2 je jedna z nasledujúcich rezolvent:

(1) binárna rezolventa C_1 a C_2
(2) binárna rezolventa C_1 a spojenia C_2
(3) binárna rezolventa spojenia C_1 a C_2
(4) binárna rezolventa spojenia C_1 a spojenia C_2

Poznámka 3.29. Sú možné aj ohraničenia na spojenia.

Príklad 3.30. Nech $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ a $C_2 = \neg P(f(g(a))) \vee Q(b)$. Spojenie C_1 je $C_1' = P(f(y)) \vee R(g(y))$. Binárna rezolventa C_1' a C_2 je $R(g(g(a))) \vee Q(b)$. Z toho vyplýva, že $R(g(g(a))) \vee Q(b)$ je rezolventa C_1 a C_2.

Pravidlo rezolvent je odvodzovacie pravidlo, ktoré indukuje rezolventy na množine klauzúl. Toto pravidlo v roku 1965 zaviedol Robinson. Je efektívnejšie ako predchádzajúce metódy dôkazov, napr. ako priama aplikácia Herbrandovej vety, ktorú použil Gilmore a neskôr Davis a Putman. Okrem toho, metóda rezolvent je úplná, t.j. pri pomoci pravdla rezu môžeme pre ľubovoľnú nesplniteľnú množinu získať prázdnu klauzulu $\square$. V ďalšom dokážeme uvedené tvrdenie.

Poznamenávame, že ak posledná odvodená klauzula metódou rezolvent je prázdna, tak urobíme záver, že množina klauzúl S nie je splniteľná.

Kroky v dôkaze môžeme ľahko vyjadriť stromom. Strom nazývame *stromom odvodenia*, t.j. strom odvodenia z množiny S je hore rastúci strom, pričom každému jeho visiacemu vrcholu pripíšeme klauzulu z S a každému nasledujúcemu vrcholu pripisujeme rezolventu vrcholov (klauzúl) bezprostredne predchádzajúcich vrcholu. Strom odvodenia nazývame *stromom odvodenia klauzuly* R, ak je R pripísaná koreňu stromu. Strom odvodenia je prosto strom, ktorý vyjadruje odvodenie. V dôsledku toho budeme používať termíny „odvodenie" a „strom odvodenia" ako zameniteľné.

3.5. Úplnosť metódy rezolvent

Pri dôkaze Herbrandovej vety sme zaviedli pojem sémantického stromu. V tejto časti budeme používať sémantický strom na dôkaz úplnosti metódy rezolvent. Skutočne, existuje blízka súvislosť medzi sémantickým stromom a odvodením pomocou rezolvent, čo demonštrujeme nasledujúcim príkladom:

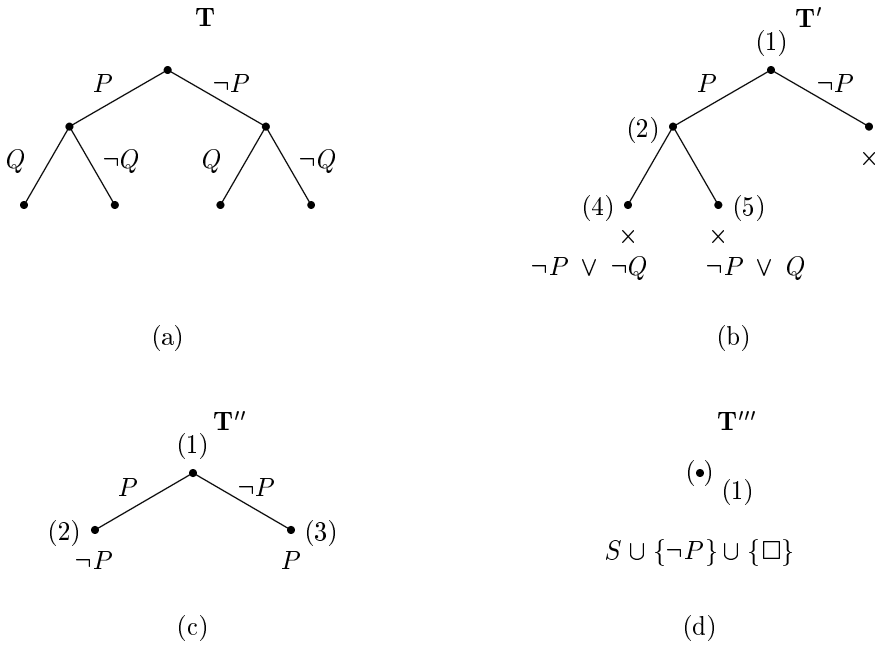
Príklad 3.31. Uvažujme nasledujúcu množinu klauzúl S:

(1) P
(2) $\neg P \vee Q$
(3) $\neg P \vee \neg Q$

Herbrandovská báza S je $\{P, Q\}$. Nech T je uzavretý sémantický strom na obr. 3.3(a). T má uzavretý sémantický podstrom T' na obr. 3.3(b). Uzol (2) na obr. 3.3(b) je akceptujúcim vrcholom. No dva jeho nasledovníky (4) a (5) sú odmietajúce vrcholy. Klauzuly, ktoré zodpovedajú vrcholom (4) a (5) budú $\neg P \vee \neg Q$ a $\neg P \vee Q$ v uvedenom poradí. Ľahko vidno, že tieto dve klauzuly musia mať kontrárnu dvojicu literálov a z toho vyplýva, že môžu byť predpokladmi pravidla rezu. Ak spojíme $\neg P \vee \neg Q$ a $\neg P \vee Q$, dostávame $\neg P$. Poznamenávame, že $\neg P$ sa odmieta čiastočnou interpretáciou, ktorá zodpovedá vrcholu (2). Ak pridáme k S klauzulu $\neg P$, tak budeme mať uzavretý sémantický strom T'' pre $S \cup \{\neg P\}$, zobrazený na obr. 3.3(c)., kde vrchol (1) je akceptujúci vrchol. Súčasne môže vzniknúť $\square$, a to aplikovaním pravidla rezu k P a $\neg P$. Ak pridáme $\square$ do $S \cup \{\neg P\}$, dostaneme uzavretý sémantický strom T''' pre $S \cup \{\neg P\} \cup \{\square\}$, zobrazený na obr. 3.3(d). Opísané „sťahovanie" sémantického stromu v skutočnosti zodpovedá nasledujúcemu rezolvenčnému odvodeniu pre množinu $S = \{P, \neg P \vee Q, \neg P \vee \neg Q\}$:

(4) $\neg P$ — rezolventa (2) a (3)
(5) $\square$ — rezolventa (4) a (1)

V ďalšom budeme používať uvedenú ideu, aby sme dokázali úplnosť metódy rezolvent,t.j. zostrojíme uzavretý sémantický strom pre nesplniteľnú množinu klauzúl a postupne spolu s uskutočňovaním metódy rezolvent „sťahujeme" strom do jedného vrchola. Skôr než dokážeme vetu o úplnosti, dokážeme pomocné tvrdenie.



Obrázok 3.3.

Lema 3.32. Nech C_1' a C_2' sú inštancie C_1 a C_2 v uvedenom poradí. Ak C' je rezolventa C_1' a C_2', tak existuje taká rezolventa C klauzúl C_1 a C_2, že C' je inštancia C.

Dôkaz: ▷ Ak treba, tak premenujeme premenné v C_1 a C_2 tak, aby C_1 a C_2 nemali spoločné premenné. Nech L_1' a L_2' sú literály, ktoré môžeme vynechať a nech $C' = (C_1'\nu - L_1'\nu) \cup (C_2'\nu - L_2'\nu)$, kde ν je najvšeobecnejší unifikátor L_1' a $\neg L_2'$. C_1' a C_2' sú inštancie C_1 a C_2 v uvedenom poradí. Preto existuje taká substitúcia θ, že $C_1' = C_1\theta$ a $C_2' = C_2\theta$. Nech $L_i^1, \ldots, L_i^{r_i}$ sú literály v C_i — zodpovedajúce L_i', t.j. $L_i^1\theta = \cdots = L_i^{r_i}\theta = L_i'$ $(i = 1, 2)$. Ak $r_i > 1$, dostávame najvšeobecnejší unifikátor λ_i pre $\{L_i^1, \ldots, L_i^{r_i}\}$. Nech $L_i = L_i^1\lambda_i$ $(i = 1, 2)$. Pretože λ_i je najvšeobecnejší unifikátor, tak pre vhodnú substitúciu ξ platí

$$L_i' = L_i^1\theta = L_i^1(\lambda_i \circ \xi) = (L_i^1\lambda_i)\xi = L_i\xi$$

teda $L_i\xi = L_i'$. L_i je pritom literál v spojení $C_i\lambda_i$ pre C_i. Ak $r_i = 1$, tak $\lambda_i = \varepsilon$ a $L_i = L_i^1\lambda_i$. Nech $\lambda = \lambda_1 \cup \lambda_2$. Tak je zrejmé, že L_i' je inštancia L_i.

Pretože L_1' a $\neg L_2'$ sú unifikovateľné, tak aj L_1 a $\neg L_2$ sú unifikovateľné. Nech σ je najvšeobecnejší unifikátor pre L_1 a $\neg L_2'$. Nech

$$\begin{aligned} C &= ((C_1\lambda)\sigma - L_1\sigma) \cup ((C_2\lambda)\sigma - L_2\sigma) = \\ &= ((C_1\lambda)\sigma - (\{L_1^1, \ldots, L_1^{r_1}\}\lambda)\sigma) \cup ((C_2\lambda)\sigma - (\{L_2^1, \ldots, L_2^{r_2}\}\lambda)\sigma) = \\ &= (C_1(\lambda \circ \sigma) - \{L_1^1, \ldots, L_1^{r_1}\}(\lambda \circ \sigma)) \cup (C_2(\lambda \circ \sigma) - \{L_2^1, \ldots, L_2^{r_2}\}(\lambda \circ \sigma)) \end{aligned}$$

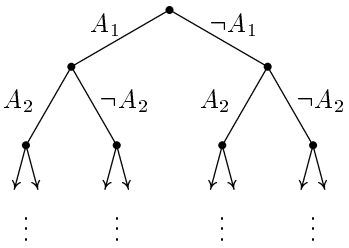
C je rezolventa C_1 a C_2. Je zrejmé, že C' je inštancia C, pretože

$$\begin{aligned} C' &= (C_1'\nu - L_1'\nu) \cup (C_2'\nu - L_2'\nu) = \\ &= ((C_1\theta)\nu - (\{L_1^1, \ldots, L_1^{r_1}\}\theta)\nu) \cup ((C_2\theta)\nu - (\{L_2^1, \ldots, L_2^{r_2}\}\theta)\nu) = \\ &= (C_1(\theta \circ \nu) - \{L_1^1, \ldots, L_1^{r_1}\}(\theta \circ \nu)) \cup (C_2(\theta \circ \nu) - \{L_2^1, \ldots, L_2^{r_2}\}(\theta \circ \nu)) \end{aligned}$$

a $\lambda \circ \sigma$ je všeobecnejší ako $\theta \circ \nu$, pretože λ je všeobecnejší ako θ a σ je všobecnejší ako ν. Tým sme dokázali lemu. ◁

Veta 3.33 (Úplnosť rezolvenčnej metódy). Množina klauzúl S nie je splniteľná práve vtedy, keď existuje odvodenie prázdnej klauzuly $\square$ z S.

Dôkaz: $\triangleright$ ($\Longrightarrow$) Predpokladajme, že S nie je splniteľná. Nech $A = \{A_1, A_2, A_3 \dots\}$ je množina atómov S. Nech T je uzavretý sémantický strom uvedený na obr. 3.4.



Obrázok 3.4.

Podľa Herbrandovej vety (I. variant) T obsahuje konečný uzavretý sémantický strom T'. Ak sa T' skladá len z jedného vrchola (koreňa), tak $\square$ musí patriť do S, pretože žiadna iná klauzula nemôže byť odmietnutá v koreni sémantického stromu. Je zrejmé, že v tom prípade nám veta platí. Predpokladajme, že sa T' skladá z viacej ako jedného vrchola. Potom T' má aspoň jeden akceptujúci vrchol. Keby to tak totiž nebolo, tak by mal každý vrchol ako potomka (nasledovníka) aspoň jeden neodmietajúci vrchol. V tom prípade by sme však mohli nájsť nekonečnú vetvu vychádzajúcu z T', čo je v spore s konečnosťou T'.

Nech v je akceptujúci vrchol v T' a nech v_1 a v_2 sú odmietajúce vrcholy, ktoré ležia bezprostredne nižšie. Nech $I(v) = \{m_1, m_2, \dots, m_n\}$, $I(v_1) = \{m_1, m_2, \dots, m_n, m_{n+1}\}$, $I(v_2) = \{m_1, m_2, \dots, m_n, \neg m_{n+1}\}$. Pretože v_1 a v_2 sú odmietajúce vrcholy a v neodmietajúci vrchol, tak musia existovať dve základné inštancie C_1' a C_2' klauzúl C_1 a C_2 také, že C_1' a C_2' neplatia v $I(v_1)$ a $I(v_2)$ v uvedenom poradí, no C_1' a C_2' sa nezamietajú $I(v)$. Z toho vyplýva, že C_1' musí obsahovať $\neg m_{n+1}$ a C_2' musí obsahovať m_{n+1}. Nech $L_1' = \neg m_{n+1}$ a $L_2' = m_{n+1}$. Ak vynecháme literály L_1' a L_2', môžeme dostať rezolventu C' pre C_1' a C_2', a to práve je $C' = (C_1' - L_1') \cup (C_2' - L_2')$. C' musí byť nepravdivá v $I(v)$, pretože $(C_1' - L_1')$ a $(C_2' - L_2')$ neplatia v $I(v)$. Podľa predchádzajúcej lemy existuje taká rezolventa C z C_1 a C_2, že C' je základná inštancia C. Nech T'' je uzavretý sémantický strom pre $(S \cup \{C\})$, ktorý dostaneme z T' vynechaním ľubovoľného vrchola alebo hrany, ktorá sa nachádza nižšie než prvý vrchol, v ktorom sa rezolventa C' odmieta. Je zrejmé, že počet vrcholov v T'' je menší ako počet vrcholov v T'. Ak aplikujeme vyššie uvedený postup opäť na T'', dostávame ďalšiu rezolventu v $(S \cup \{C\})$ a môžeme dostať iný sémantický strom s menším počtom vrcholov. Tento postup opakujeme dovtedy, pokým nevznikne uzavretý sémantický strom, ktorý sa skladá z len z koreňového vrchola. To je možné len vtedy, ak je odvodená $\square$. Z toho vyplýva, že existuje odvodenie $\square$ z S.

($\Longleftarrow$) Obrátene, predpokldajme, že existuje odvodenie $\square$ z S. Nech $R_1, R_2, \dots, R_k$ sú rezolventy v odvodení. Predpokladajme, že S je splniteľná na modeli M. No ak model vyhovuje klauzulám C_1 a C_2, tak musí vyhovovať aj ľubovoľnej ich rezolvente. Z toho vyplýva, že M vyhovuje klauzulám $R_1, R_2, \dots, R_k$. No posledné tvrdenie nemôže platiť, pretože jedna z uvažovaných relovent je $\square$. Preto S musí byť nesplniteľná, čo sme mali dokázať. $\triangleleft$

Príklad 3.34. Uvažujme nasledujúcu množinu formúl:

$$F_1\colon\ (\forall x)\big(C(x) \to (W(x) \ \wedge\ R(x))\big)$$
$$F_2\colon\ (\exists x)\big(C(x) \ \wedge\ Q(x)\big)$$
$$G\colon\ (\exists x)\big(Q(x) \ \wedge\ R(x)\big)$$

Našou úlohou je dokázať, že G je logickým dôsledkom F_1 a F_2.

Riešenie: Vytvoríme pre F_1, F_2 a $\neg G$ štandardnú formu a dostaneme nasledujúcich 5 klauzúl

(1) $\neg C(x) \vee W(x)$ z F_1
(2) $\neg C(x) \vee R(x)$ z F_1
(3) $C(a)$ z F_2
(4) $Q(a)$ z F_2
(5) $\neg Q(x) \vee \neg R(x)$ z $\neg G$

Táto množina klauzúl nie je splniteľná. Môžeme to dokázať pomocou metódy rezolvent nasledujúcim spôsobom.

(6) $R(a)$ — rezolventa (3) a (2)
(7) $\neg R(a)$ — rezolventa (5) a (4)
(8) $\square$ — rezolventa (7) a (6)

Preto je G logickým dôsledkom F_1 a F_2.

3.6. Stratégia vymazávania

V predchádzajúcej časti sme dokázali úplnosť metódy rezolvent. Táto metóda je efektívnejšia ako metódy, ktoré sa používali predtým. No nie príliš rozvážne aplikovanie pravidla rezu môže indukovať veľké množstvo zbytočných klauzúl. Na to, aby sme sa o tom presvedčili, uvedieme jednoduchý príklad:

Majme množinu klauzúl $S = \{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$. Metódou rezolvent chceme ukázať, že množina S nie je splniteľná.

Aplikácia metódy rezolvent pre množinu S spočíva vo vyčíslení všetkých rezolvent všetkých dvojíc klauzúl S, pridaní týchto rezolvent k množine S, určení všetkých ďalších rezolvent a v opakovaní tohto procesu dotiaľ, pokým nedostaneme prázdnu klauzulu $\square$. To znamená, že tvoríme sekvencie $S^0, S^1, S^2, \ldots$, kde

$$S^0 = S$$
$$S^n = \left\{\text{rezolventy } C_1 \text{ a } C_2 \mid C_1 \in \left(S^0 \cup \cdots \cup S^{n-1}\right) \wedge C_2 \in S_{n-1}\right\} \quad n = 1, 2, \ldots$$

Táto metóda sa nazýva *metóda nasýtenia úrovne*. Inými slovami, postupujeme nasledovne: Najprv zapíšeme klauzuly $\left(S^0 \cup \cdots \cup S^{n-1}\right)$ v istom poradí a potom vyčíslime rezolventy porovnávajúc každú klauzulu $C_1 \in \left(S^0 \cup \cdots \cup S^{n-1}\right)$ s klauzulou $C_2 \in S^{n-1}$, ktorá sa nachádza po C_1. Keď utvoríme rezolventu, pripíšeme ju na koniec zoznamu, ktorý bol dovtedy vytvorený. Ak použijeme uvedenú metódu na množinu klauzúl S z príkladu, zostrojíme sekvencie $S^0, S^1, S^2, \ldots$, obsahujúce 38 klauzúl a ako 39. sa objaví prázdna klauzula $\square$.

Vytvorili sme veľa klauzúl, ktoré v našom prípade nepotrebujeme, t.j. sú nadbytočné. Môžu to byť napríklad tautológie. Pretože tautológia je pravdivá v ľubovoľnej interpretácii, tak ak ju vynecháme z nejakej nesplniteľnej množiny klauzúl, množina zostávajúcich klauzúl je nesplniteľná. Z toho vyplýva, že tautológia nemá vyplyv na výsledok, a teda nie je potrebné ju vytvárať. Ak ju teda vytvoríme, tak ju treba vynechať. V opačnom prípade môže dávať s inými klauzulami nadbytočné klauzuly (jedna a tá istá klauzula vznikne viackrát). Ďalej môžu vznikať niektoré klauzuly viackrát, aj keď nepoužijeme tautológiu. Vzniká nám teda veľa zbytočností. Na riešenie nadbytočnosti rozoberieme v ďalšom stratégiu vymazávania.

Definícia 3.35. Klauzula C je *podklauzulou* D (alebo *pohlcuje* D) práve vtedy, keď existuje taká substitúcia σ, že $C\sigma \subseteq D$. D nazývame *nadklauzulou* C.

Príklad 3.36. Nech $C = P(x)$ a $D = P(a) \vee Q(a)$. Ak $\sigma = \{a/x\}$, tak $C\sigma = P(a)$. Pretože $C\sigma \subseteq D$, tak C je podklauzula D.

Poznamenávame, že ak D je identické C alebo D je inštancia C, tak D je nadklauzula C. Stratégia vymazávania spočíva vo vynechávaní ľubovoľných tautológií a nadklauzúl, kde je to možné. Úplnosť vymazávania závisí od toho, ako sa vynechávajú tautológie a nadklauzuly.

Robíme to nasledujúcim spôsobom (používame ju spolu s metódou nasýtenia úrovne): Najprv vypisujeme klauzuly $\left(S^0 \cup \cdots \cup S^{n-1}\right)$ v istom poradí. Potom vypisujeme rezolventy tak, že porovnávame každú klauzulu $C_1 \in \left(S^0 \cup \cdots \cup S^{n-1}\right)$ s klauzulou $C_2 \in S^{n-1}$, ktorá je zapísaná po C_1. Keď získame rezolventu, tak ju zapisujeme na koniec zoznamu, ak nie je tautológia a nie je pohltená žiadnou klauzulou zo zoznamu. V opačnom prípade ju vynechávame.

Príklad 3.37. Príklad na použitie tohto postupu je na obr. 3.5.

$S = S^0$:	(1)	$P \vee Q$	
	(2)	$\neg P \vee Q$	
	(3)	$P \vee \neg Q$	
	(4)	$\neg P \vee \neg Q$	
S^1:	(5)	Q	z (1) a (2)
	(6)	P	z (1) a (3)
	(7)	$\neg P$	z (2) a (4)
	(8)	$\neg Q$	z (1) a (4)
S^2:	(9)	$\square$	z (5) a (8)

Obrázok 3.5.

Poznamenávame, že tento zoznam je omnoho kratší ako zoznam, ktorý sme vytvorili predtým. Z toho vyplýva, že stratégia vymazávania môže zlepšiť efektívnosť metódy rezolvent.

Aby sme mohli použiť stratégiu vymazávania, musíme vedieť riešiť otázku, či je klauzula tautológia alebo či je jedna z klauzúl podklauzulou druhej. Ľahšie sa určuje, či je klauzula tautológia — stačí preveriť výskyt kontrárnych dvojíc. No preverenie podklauzúl nie je také jednoduché. Opíšeme algoritmus preverenia vlastnosti „byť podklauzulou".

Nech C a D sú klauzuly. Nech $\theta\{a_1/x_1, \ldots, a_n/x_n\}$, kde $x_1, \ldots, x_n$ sú premenné, ktoré sa vyskytujú v D a $a_1, \ldots, a_n$ sú nové rôzne konštanty, ktoré sa nevyskytujú v C alebo D. Položíme $D = L_1 \vee L_2 \vee \cdots \vee L_m$. Potom $D\theta = L_1\theta \vee L_2\theta \vee \cdots \vee L_m\theta$. Poznamenávame, že $D\theta$ je základná klauzula. $\neg D\theta = \neg L_1\theta \wedge \cdots \wedge \neg L_m\theta$. Nasledujúci algoritmus preveruje, či je C podklauzulou D.

Algortimus pohltenia

Krok 1. Nech $W = \{\neg L_1\theta, \ldots, \neg L_m\theta\}$.

Krok 2. Kladieme $k = 0$ a $U^0 = \{C\}$

Krok 3. Ak U^k obsahuje $\square$, tak koniec: C je podklauzula D. V opačnom prípade kladieme $U^{k+1} = \{\text{rezolventa } C_1 \text{ a } C_2 \mid C_1 \in U^k \wedge C_2 \in W\}$.

Krok 4. Ak U^{k+1} je prázdna množina $\emptyset$, tak koniec: C nie je podklauzula D. V opačnom prípade kladieme $k = k + 1$ a prejdeme ku kroku 3.

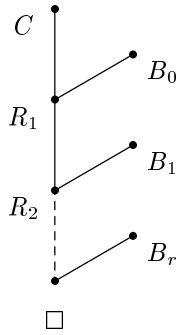
Obrázok 3.6. Algoritmus pohltenia

Poznamenávame, že v tomto algoritme je každá klauzula v U^{k+1} o jeden literál kratšia ako klauzula v U^k, z ktorej sme ju dostali. Preto sa v postupnosti $U^0, U^1, \ldots$ musí vyskytnúť množina, ktorá obsahuje $\square$ alebo prázdna množina. Algoritmus pohltenia je korektný, t.j. C je podklauzula D práve vtedy, keď algoritmus zakončuje prácu na 3. kroku. To možno dokázať nasledujúcim spôsobom.

Dôkaz: ▷ (1) Ak C je podklauzula D , tak existuje taká substitúcia σ, že $C\sigma \subseteq D$. Z toho vyplýva, že $C(\sigma \circ \theta) \subseteq D\theta$. Takým spôsobom literály $C(\sigma \circ \theta)$ môžeme vynechať použitím jednotkových

základných klauzúl vo W. $C(\sigma \circ \theta)$ je inštancia C. Z toho vyplýva, že literály v C môžeme vynechať použitím jednotkových klauzúl vo W. To znamená, že nakoniec nájdeme U^k, obsahujúcu $\square$. Preto algoritmus zakončuje prácu na 3. kroku.

(2) Obrátene, ak algoritmus zakončuje prácu na 3. kroku, tak dostávame odmietnutie, ako na obr. 3.7, kde $B_0, \ldots, B_k$ sú klauzuly z W, R_1 je rezolventa C a B_0 a R_i je rezolventa R_{i-1} a B_{i-1} pre $i = 2, \ldots, r$. Potom $C(\sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_r) = \{\neg B_0, \neg B_1, \ldots, \neg B_r\} \subseteq D\theta$.



Obrázok 3.7.

Nech $\lambda = \sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_r$. Potom $C\lambda \subseteq D\theta$. Nech σ je substitúcia, ktorú dostaneme z λ zámenou v každom komponente λ, a to tak, že a_i zameníme x_i pre $i = 1, \ldots, n$. Potom $C\sigma \subseteq D$ Z toho vyplýva, že C je podklauzula D, čo bolo treba dokázať.

Príklad 3.38. Nech $C = \neg P(x) \vee Q(f(x), a)$ a $D = \neg P(h(y)) \vee Q(f(h(y)), a) \vee \neg P(z)$. Zistite, či C je podklauzula D.

Riešenie:

(1) y a z sú premenné v D. Nech $\theta = \{b/y, c/z\}$. Poznamenávame, že b a c sa nevyskytujú v C a D. Potom $D\theta = \neg P(h(b)) \vee Q(f(h(b)), a) \vee \neg P(c)$. Preto $\neg D\theta = P(h(b)) \wedge \wedge \neg Q(f(h(b)), a) \wedge P(c)$. Z toho vyplýva, že

$$W = \{P(h(b)), \neg Q(f(h(b)), a), P(c)\}$$
$$U^0 = \{\neg P(x) \vee Q(f(x), a)\}$$

(2) Pretože U^0 neobsahuje $\square$, tak dostávame:

$$U^1 = \{Q(f(h(b)), a), \neg P(h(b)), Q(f(c), a)\}$$

(3) Pretože U^1 nie je prázdna a neobsahuje $\square$, tak dostaneme $U^2 = \{\square\}$.

(4) Pretože U^2 obsahuje $\square$, tak algoritmus končí svoju činnosť, môžeme teda urobiť záver, že C je podklauzula D.

Príklad 3.39. Nech $C = P(x, x)$ a $D = P(f(x), y) \vee P(y, f(x))$. Zistite, či C je podklauzula D.

Riešenie:

(1) x a y sú premenné v D. Vyberieme nové konštanty a, b rôzne od ľubovoľných konštánt v C a D. Nech $\theta = \{a/x, b/y\}$. Potom $D\theta = P(f(a), b) \vee P(b, f(a))$. $\neg D\theta = \neg P(f(a), b) \wedge \wedge \neg P(b, f(a))$. Takýmto spôsobom

$$W = \{\neg P(f(a), b), \neg P(b, f(a))\}$$
$$U^0 = \{P(x, x)\}$$

(2) Pretože U^0 neobsahuje $\square$, tak dostaneme

$$U^1 = \emptyset$$

(3) Pretože U^1 je prázdna, tak algoritmus končí svoju činnosť a môžeme urobiť záver, že C nie je podklauzula D.

3.7. Niektoré príklady na použitie metódy rezolvent

Príklad 3.40. Majme formuly

(1) $P \to S$
(2) $S \to U$
(3) P
(4) U

Máme dokázať, že (4) vyplýva z (1), (2) a (3).

Riešenie: Najskôr vyjadríme všetky tvrdenia v štandardnej forme. Takým spôsobom dostávame

(1') $\neg P \vee S$
(2') $\neg S \vee U$
(3') P
(4') U

Zamietnutím dokážeme, že U je logický dôsledok z (1'), (2'), (3'). Urobíme negáciu (4') a dostávame nasledujúci dôkaz:

(1) $\neg P \vee S$
(2) $\neg S \vee U$
(3) P
(4) $\neg U$ — negácia záveru
(5) S — rezolventa (3) a (1)
(6) U — rezolventa (5) a (2)
(7) $\square$ — rezolventa (6) a (4)

Príklad 3.41.

- **Predpoklad:** Študenti sú občania.
- **Záver:** Hlasy študentov sú hlasy občanov.

Riešenie: Nech

- $S(x)$ označuje „x je študent"
- $C(x)$ označuje „x je občan"
- $V(x, y)$ znamená „x je hlas y"

Takým spôsobom môžeme napísať:

- **Predpoklad:** $(\forall y)\big(S(y) \to C(y)\big)$
- **Záver:** $(\forall x)\big((\exists y)(S(y) \wedge V(x,y)) \to (\exists y)(C(z) \wedge V(x,z))\big)$

Ľubovoľný hlas môžeme priradiť študentovi alebo inému občanovi. Štandardná forma predpokladu je:

(1) $\neg S(y) \vee C(y)$

Ďalej, pretože

$$\begin{aligned}
&\neg\Big((\forall x)\big((\exists y)(S(y) \wedge V(x,y)) \to (\exists y)(C(z) \wedge V(x,z))\big)\Big) = \\
&= \neg\Big((\forall x)\big((\forall y)(\neg S(y) \vee \neg V(x,y)) \vee (\exists z)(C(z) \wedge V(x,z))\big)\Big) = \\
&= \neg\Big((\forall x)(\forall y)(\exists z)\big(\neg S(y) \vee \neg V(x,y) \vee (C(z) \wedge V(x,z))\big)\Big) = \\
&= (\exists x)(\exists y)(\forall z)\Big(S(y) \wedge V(x,y) \wedge \big(\neg C(z) \vee \neg V(x,z)\big)\Big)
\end{aligned}$$

dostávame tri klauzuly pre negáciu záveru:

(2) $S(b)$

(3) $V(a, b)$
(4) $\neg C(z) \vee \neg V(a, z)$

Dôkaz zakončujeme nasledujúcim spôsobom:

(5) $C(b)$ — z (1) a (2)
(6) $\neg V(a, b)$ — z (4) a (5)
(7) $\square$ — z (3) a (6)

Predpokladajme, že b je študent, a je hlas študenta b, a nie je hlas žiadneho občana. Pretože b je študent, b je občan. Okrem toho, a nemôže byť hlas b, pretože b je občan. A to nie je možné.